ALGORITMA & PEMROGRAMAN

Humuntal Rumapea
Darwis R. Manalu

USU press

ALGORITMA
&
PEMROGRAMAN C++

HUMUNTAL RUMAPEA
DARWIS R. MANALU

USU press
2013

KATA PENGANTAR

Puji syukur Penulis panjatkan kepada Tuhan Yang Maha Kuasa karena dapat menulis buku ini hingga selesai dengan baik. Materi yang disajikan ini adalah salah satu kebutuhan pemrograman dasar yang harus dipelajari saat ini yaitu algoritma pemrograman sebagai langkah awal dalam membuat kode program yang ditujukan bagi pembuat program komputer (*programmer*). Algoritma dan Pemrograman pada buku ini menggunakan Bahasa Pemrograman ini yang sangat tepat digunakan untuk mengimplementasikan logika dan algoritma menjadi sebuah program secara modular maupun berbasis objek serta dapat digunakan untuk pemrograman bidang matematik, simulasi, antrian, grafik, permainan dan lain-lain. Pemrograman C++ mudah untuk dipelajari serta hardware yang digunakan tidaklah harus canggih tetapi dengan kemampuan komputer processor dan memori yang lebih rendah dapat menggunakannya dengan baik. Pemrograman ini dapat mendukung pemrograman berorintasi objek (*Object Oriented Programming*)

Buku ini ditulis dengan sistematis yang dimulai dari pemahaman tentang Algoritma, jenis algoritma, penerapan algoritma dan pengenalan bahasa pemrograman, struktur penulisan, tipe data, operator, statement input/output, seleksi/pencabangan, perulangan/iterasi, struktur, pointer, array, fungsi string serta pengolan input/output data (file), structur, class dan object, pengurutan, pencarian data. Setiap bab pada buku ini dilengkapi dengan beberapa contoh dan latihan pemrograman.

Setelah membaca dan mencoba contoh-contoh program yang ada pada buku ini, diharapkan Pembaca dapat menguasai Algoritma dan pemrograman, serta mampu mengaplikasikannya pada kasus-kasus pemrograman. Harapan penulis Pembaca nantinya mampu mengembangkannya secara mandiri. Akhir kata penulis mengucapkan terimakasih, selamat belajar dan semoga buku ini bermanfaat.

Medan, Januari 2013

Penulis

DAFTAR ISI

BAB 1, ALGORITMA DAN BAHASA PEMROGRAMAN

1.1 Pengertian Algoritma 1
1.2 Bahasa Pemrograman 4
1.3 Dasar-dasar Algoritma 6
1.4 Struktur Dasar Algoritma 7
1.5 Penyajian Algoritma 8
1.6 Tujuan Pemrograman 13
1.7 Manfaat Pemrograman 13
1.8 Tahapan Dalam Pemrograman 13

BAB 2, BAHASA PEMROGRAMAN C++

2.1 Bahasa C++ 15
2.2 Struktur Bahasa Program 15
2.3 Sejarah Singkat C++ 17
2.4 Tampilan Editor Program Bahasa C++ 19
2.5 Cara Penggunaan Menu 20
2.6 Contoh Penulisan Program 21
2.7 Keyword Dalam C++ 23

BAB 3, ATURAN PENULISAN PROGRAM C++

3.1 Susunan Program C++ 24
3.2 Fungsi Utama 25
3.3 Include File Header 25

BAB 4, JENIS DATA, VARIABEL & OPERATOR

4.1 Kata Kunci 30
4.2 Pengenal 31
4.3 Konstanta 32
4.4 Variabel 34
4.5 Tipe Data 35

BAB 5, OPERATOR DAN EKSPRESI

5.1 Operator Aritmatika 40
5.2 Operator Relasional 41
5.3 Operator Logika 41
5.4 Operator Sizeof 41
5.5 Operator Penugasan 43
5.6 Operator Koma 43

5.7 Operator Bertingkat 44
5.8 Operator Alamat 44
5.9 Operator Bersyarat 44
5.10 Ekspresi 45

BAB 6, FUNGSI INPUT DAN OUTPUT

6.1 Fungsi Output 47
6.2 Fungsi Input 51

BAB 7, SELEKSI DAN PECABANGAN

7.1 Struktur if 56
7.2 Struktur if-else 58
7.3 Struktur else – if 59
7.4 Struktur switch 65
7.5 Struktur goto 72

BAB 8, PERULANGAN

8.1 Pengulangan for 76
8.2 Pengulangan while 79
8.3 Pengulangan Bertingkat 82

BAB 9, POINTER

9.1 Mendekarasikan Variabel Pointer 97
9.2 Inisialisasi Variabel Pointer 98
9.3 Operator Indireksi dan Operator Alamat 99

BAB 10, FUNGSI

10.1 Kategori Fungsi 103
10.2 Prototipe Fungsi 104
10.3 Pemanggilan Fungsi 104
10.4 Mengembalikan Nilai Fungsi 106

BAB 11, LARIK (ARRAY)

11.1 Array Satu Dimensi 118
11.2 Array Dua Dimensi 122

BAB 12, FUNGSI STRING

12.1 Menyalin String 126
12.2 Menggabungkan String 127
12.3 Menghitung Panjang String 127
12.4 Mengubah Karakter 128

BAB 13, PENGURUTAN DATA (SORTING)

13.1 Defenisi Pengurutan 136
13.2 Pengurutan Internal dan Pengurutan Eksternal 136
13.3 Algoritma Pengurutan Gelembung (Buble Sort) 137
13.4 Pengurutan Pilih 138
13.5. Pengurutan Sisip 145
13.6. Penggabungan Dua Buah Larik Terurut 148

BAB 14, PENCARIAN (SEARCHING)

14.1 Tinjauan Singkat Larik 149
14.2 Defenisi Persoalan Pencarian di dalam Larik 149
14.3 Pencarian Beruntun [Sequential Search] 150
14.4 Pencarian Bagidua [Binary Search] 151

BAB 15, STRUKTUR DAN FILE

15.1 Mendeklarasikan Struktur 162
15.2 Inisialisasi Struktur 164
15.3 Mengakses Anggota Struktur 164
15.4 Array Struktur 165
15.5 File 170
15.6 Membuka dan menutup File 170
15.7 Model-model Akses File..... 171
15.8 Operasi Input/output File 171

BAB 16, CLASS DAN OBJECT

16.1 Data Member 174
16.2 Member Function 175

BAB 17, LATIHAN

17.1 Contoh Membuat Grafik 188
17.2 Contoh Bintang Segi Enam 190
17.3 Contoh Bilangan Random..... 191
17.4 Contoh Pengurutan Data 192
17.5 Contoh Menampilkan Bintang 193
17.6 Contoh Progam Game Ular 195
Contoh Rekursif Faktorial..... 202
Contoh Program Rekursif..... 203
Contoh Program untuk Kode ASCII 204

DAFTAR PUSTAKA 208

BAB 1 ALGORITMA DAN BAHASA PEMROGRAMAN

1.1 Pengertian Algoritma

Perkataan Algoritma memiliki sejarah tersendiri karena adanya perubahan kata dari asalnya hingga menjadi popular saat ini. Para pembuat program komputer maupun penggunanya menemukan kata algorism yang berarti proses menghitung dengan angka arab. Pengguna computer dikatakan *algorist* jika dapat menghitung dengan menggunakan angka arab. Para ahli bahasa berusaha menemukan asal kata ini namun hasilnya kurang memuaskan. Akhirnya para ahli sejarah matematika menemukan asal kata tersebut yang berasal dari nama penulis buku arab yang terkenal yaitu Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi. Al-Khuwarizmi dibaca orang barat menjadi Algorism. Al-Khuwarizmi menulis buku yang berjudul Kitab Al Jabar Wal-Muqabala yang artinya "Buku pemugaran dan pengurangan" (*The book of restoration and reduction*). Dari judul buku tersebut diperoleh akar kata "Aljabar" (Algebra).

Perubahan kata dari algorism menjadi algorithm muncul karena kata algorism sering dikelirukan dengan arithmetic, sehingga akhiran –sm berubah menjadi –thm. Karena perhitungan dengan angka Arab sudah menjadi hal yang biasa, maka seiring dengan perkembangan teknologi informasi lambat laun kata algorithm berangsur-angsur dipakai sebagai metode perhitungan (*komputasi*) secara umum, sehingga kehilangan makna kata aslinya. Dalam bahasa Indonesia saat ini bahkan dalam penggunaan, kata algorithm diserap menjadi algoritma.

Defenisi :

> **Algoritma** adalah tahapan atau urutan langkah-langkah logik penyelesaian masalah yang disusun secara sistematis.

Pembuatan dan penggunaan algoritma sangat erat hubunganya dengan pembuat progam (*programmer)*. Karena dalam tahap inilah dapat ditunjukkan bagan secara global permasalahan serta mendeskripsikan berbagai tugas dan fungsi masing-masing bagian yang ada. Algoritma adalah hal yang terpenting dilakukan dalam membuat suatu program. Karena berdasarkan algoritma inilah seorang programmer dapat menerjemahkan ke dalam bahasa pemrograman (*coding*). Algoritma merupakan gambaran/urutan pengerjaan atau penyelesaian suatu masalah secara lengkap dan teratur.

Menurut Donald E. Knuth algoritma tersebut adalah :

- Algoritma mempunyai awal dan akhir.
- Setiap langkah harus didefinisikan dengan tepat sehingga tidak memiliki arti ganda (*ambigu*).
- Selalu memiliki masukan (*input*) atau kondisi awal.
- Dan memiliki luaran (output) atau kondisi akhir.
- Kemudian algoritma juga harus efektif; bila digunakan benar–benar menyelesaikan persoalan.

Dalam kenyataannya pada saat menyelesaikan permasalahan tentu saja sering terjadi adanya pembatasan, meskipun terkadang terkesan terlalu membatasi. Prakteknya penyelesaian masalah sebenarnya ada banyak cara untuk menghasilkan sesuatu tanpa harus mengikuti aturan yang "baku" / satu aturan saja yang terkesan mengikat,. Namun jika ada yang mau bertindak seperti yang telah dijelaskan diatas tentu saja dapat diterima dengan syarat bersifat logis dan hasil luaran yang jelas.

Dalam beberapa konteks, algoritma adalah spesifikasi urutan langkah untuk melakukan pekerjaan tertentu. Pertimbangan dalam pemilihan algoritma adalah:

pertama, algoritma haruslah benar. Artinya algoritma akan memberikan keluaran yang dikehendaki dari sejumlah masukan yang diberikan. tidak peduli sebagus apapun algoritma, kalau memberikan keluaran yang salah,pastilah algoritma tersebut bukanlah algoritma yang baik.

Kedua, yang harus diperhatikan adalah kita harus mengetahui seberapa baik hasil yang dicapai oleh algoritma tersebut. Hal ini penting terutama pada algoritma untuk menyelesaikan masalah yang memerlukan aproksimasi hasil (*hasil yang hanya berupa pendekatan*). Algoritma yang baik harus mampu memberikan hasil yang sedekat mungkin dengan nilai yang sebenarnya.

Ketiga adalah efisiensi algoritma. Efisiensi algoritma dapat ditinjau dari 2 hal yaitu efisiensi waktu dan memori. Meskipun algoritma memberikan keluaran yang benar (*paling mendekati*), tetapi jika harus menunggu berjam-jam untuk mendapatkan keluarannya, algoritma tersebut biasanya tidak akan dipakai, setiap orang menginginkan keluaran yang cepat. Begitu juga dengan memori, semakin besar memori yang terpakai maka semakin buruklah algoritma tersebut.

Dalam kenyataannya, setiap orang bisa membuat algoritma yang berbeda untuk menyelesaikan suatu permasalahan, walaupun terjadi perbedaan dalam menyusun algoritma, tentunya diharapkan keluaran yang sama. Jika terjadi demikian, sehingga carilah algoritma yang paling efisien dan cepat.

Ada tiga cara penulisan algoritma yaitu:

1. Dengan menggunakan bagan atau simbol-simbol tertentu atau sering disebut dengan flowchart
2. Dengan menggunakan Pseudo Code artinya algoritma tersebut mirip dengan statemen atau perintah pada bahasa pemrograman yang digunakan
3. Dengan menggunakan kalimat sendiri.

Defenisi algotritma lainnya adalah langkah-langkah yang bersifat logis dalam penyelesaian masalah yang disusun secara sistematis. Dalam tahap inilah dapat menunjukkan bagan secara global permasalahan serta mendeskripsikan berbagai tugas dan fungsi masing-masing bagian yang ada. Algoritma adalah hal yang terpenting dilakukan dalam membuat suatu

program aplikasi. Karena berdasarkan algoritma inilah seorang programmer dapat menerjemahkan ke dalam bahasa pemrograman sesuai dengan urutannya. Algoritma merupakan gambaran/urutan pengerjaan atau penyelesaian suatu masalah secara lengkap dan teratur.

1.2 Bahasa Pemrograman

Komputer merupakan alat pengolah data elektronik yang dikontrol oleh sekumpulan instruksi dimasukkan ke dalam unit memori utama. Sekumpulan instruksi yang dikenal atau dimengerti oleh komputer dan disusun menurut urutan yang logis ini disebut program. Sedangkan bahasa yang digunakan untuk menuliskan/membuat program itu sendiri disebut dengan bahasa pemrograman (*Programming Language)* dan kegiatan pembuatan program itu disebut dengan pemrograman (*programming)* sedangkan orang yang melakukannya disebut dengan Pemrogram (*Programmer*).

Bahasa pemrograman yang digunakan untuk membuat program tentu saja harus dapat dimengerti oleh komputer. Bahasa pemrograman yang langsung dapat dimengerti oleh komputer disebut dengan bahasa mesin (*machine language*).

Dalam komputer digital yang bekerja dengan sinyal-sinyal listrik yang diskrit yaitu hasil kombinasi dari angka-angka yaitu 0 dan 1 yang artinya angka nol menyatakan tidak adanya arus atau tegangan sedangkan angka 1 menyatakan adanya arus dan tegangan sering disebut 0 adalah mati dan 1 adalah hidup. Jadi bahasa mesin pada komputer digital menggunakan sistem bilangan biner.

Bahasa pemrograman lain yang digunakan adalah bahasa mesin yang menggunakan kode singkatan perintah adalah bahasa asembly. Agar komputer dapat mengerti bahasa asembly ini diperlukan peterjemah bahasa asembly ke dalam bahasa mesin berupa program yang disebut "*assembler"* . Pemrograman dengan bahasa assembly ini masih cukup sukar dan tidak

praktis bagi kita, maka diperlukan suatu bahasa yang mudah dimengertii oleh setiap orang.

Banyak ahli yang mencoba mengembangkan suatu bahasa pemrograman yang mendekati bahasa manusia dan tidak bergantung kepada jenis komputer yang digunakan. Kemudian bahasa yang seperti ini disebut dengan bahasa pemrograman tingkat tinggi (*high level programming language)*. Contoh bahasa pemrograman tingkat tinggi sekarang banyak digunakan seperti : BASIC, TURBO PASCAL, C++, COBOL, JAVA, PROLOG, bahas Pemrogram Visual dan juga bahasa lainnya. Tentunya dengan menggunakan bahasa tersebut diatas selalu diterjemahkan terlebih dahulu ke dalam bahasa mesin agar dapat dimengerti oleh komputer itu sendiri.

Jenis penerjemah bahasa pemrograman tingkat tinggi ke dalam bahasa mesin ada dua jenis yaitu *iterpreter* dan *compiler.* Perbedaan kedua penerjemah ini adalah:

Pada *Interpreter*, perintah/instruksi dalam program sumber dibaca/diterjemahkan (*diinterpretasikan*), dan langsung dilaksanakan baris per baris progam. Jadi tidak akan memperoleh hasil terjemahan program sumber dalam bahasa mesin. Setiap kali program di jalankan atau melihat hasil (*run*), harus menggunakan program sumber dan karenanya selalu dibutuhkan interpreter untuk menerjemahkannya. Dalam arti programmer menulis programnya dengan bahasa tingkat tinggi yang mirip dengan bahasa manusia kemudian akan diterjemahkan oleh penerjemah interpreter ke dalam bahasa yang dimengerti mesin komputer.

Sedangkan *compiler* membaca keseluruhan program sumber (*Source program)* dan menerjemahkannya menjadi satu set lengkap instruksi dalam bahasa mesin, sebelum instruksi-instruksi itu dilaksanakan. Setelah proses itu dilaksanakan maka akan menghasilkan sebuah program objek (*Object Program)* dan file pendukung lainnya yang biasanya disimpan dalam media penyimpanan yang telah memiliki nama objek. Sehingga apabila ingin menjalankan program tersebut dapat menggunakan objek yang sudah dihasilkannya tanpa menggunakan program sumber file Aplikasi.

Biasanya untuk menjalankan hasil compiler dengan menuliskan nama objek (*dalam bentuk file command atau file eksekusi*) dengan ekstensi *.Com, *.Exe.

Beda Program dan Algoritma :

Program adalah kumpulan pernyataan komputer, sedangkan metode dan tahapan sistematis dalam program adalah algoritma. Program ditulis dengan menggunakan bahasa pemrograman. Jadi bisa disebut bahwa program adalah suatu implementasi dari bahasa pemrograman.

Program = Algoritma + Bahasa (*Struktur Data*)

Bagaimanapun juga struktur data dan algoritma berhubungan sangat erat pada sebuah program. Algoritma yang baik tanpa pemilihan struktur data yang tepat akan membuat program menjadi kurang baik, demikian juga sebaliknya.

1.3 Dasar-dasar Algoritma

Berpedoman pada pertemuan terdahulu, bahwa algoritma disusun oleh sederetan langkah instruksi yang logis. Tiap langkah instruksi tersebut mengerjakan suatu tindakan (aksi). Bila suatu aksi dilaksanakan, maka sejumlah operasi yang bersesuaian dengan aksi itu dikerjakan oleh pemroses.

Bentuk Umumnya :

T_o : keadaan sebelum aksi dikerjakan

Aksi

T_1 : keadaan setelah aksi dikerjakan

1.4 Struktur Dasar Algoritma

Algoritma berisi langkah-langkah penyelesaian suatu masalah. Langkah-langkah tersebut dapat berupa runtunan aksi, pemilihan aksi, dan pengulangan aksi. Sebuah algoritma dapat dibangun dari tiga buah struktur dasar, yaitu :

1. Runtunan (*sequence*)
2. Pemilihan (*selection*)
3. Pengulangan (*repetition*)

1. Runtunan

Contoh Runtunan instruksi yang dilambangkan dengan A1, A2, A3, A4, dan A5 berikut :

A1

A2

A3

A4

A5

Mula-mula pemroses melaksanakan instruksi A1. Instruksi A2 dilaksanakan setelah A1 selesai. Selanjutnya, instruksi A3 dilaksanakan setelah instruksi A2 selesai, demikian seterusnya.

2. Pemilihan

Adakalanya sebuah instruksi dikerjakan jika kondisi tertentu dipenuhi. Dalam bahasa Indonesia , **IF** berarti "jika" dan **then** artinya "maka". Kondisi adalah persyaratan yang dapat dinilai benar atau salah.

Rumus : **If** kondisi **then**

Aksi 1

Else

Aksi 2

Contoh :

If hari hujan then

Pergilah naik beca

Else

Pergilah naik motor

3. Pengulangan

Struktur pengulangan terdiri dari :

- Repeat - Until
- While - do
- For

Contoh pengulangan dengan Repeat … Until

Algoritma Menulis_FIKOM_UMI

{menulis kalimat "FIKOM_UMI" sebanyak 100 kali.}

DEKLARASI :

Kalimat : Integer; {peubah data bilangan bulat}

DESKRIPSI :

Kalimat = 0

Repeat

Tulis "FIKOM_UMI"

Naikkan kalimat dengan 1

Until = 100

1.5 Penyajian Algoritma

Ada tiga cara penulisan algoritma yaitu:

1. Naratif
2. Diagram Alir (*Flow chart*)
3. Pseudocode

1. Dengan Naratif

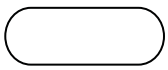
Dengan menggunakan kalimat sendiri sesuai dengan pengertian dan pemahaman seorang programmer (*untuk memudahkan programmer dalam membuat kode program*).

Contoh penulisan dengan Narasi
Dalam kegiatan bertelepon, adapun langkah-langkahnya sebagai berikut :
Langkah 1 : Mencari nomor telepon yang dituju.
Langkah 2 : Mengangkat gagang telepon.
Langkah 3 : Mendial/memutar nomor telepon.
Langkah 4 : Tunggu gagang telepon yang dituju diangkat.
Langkah 5 : Melakukan komunikasi.
Langkah 6 : Meletakkan gagang telepon.
Langkah 7 : Selesai.

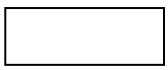
2. Dengan Flowchart

Dengan menggunakan bagan atau simbol-simbol tertentu dengan logis dan sistematis

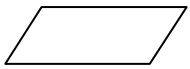
Simbol-simbol flowchart:



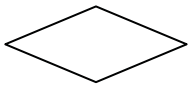
Terminator (awal dan akhir flowchart)



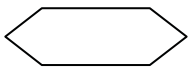
Process



Data Input dan Data Output



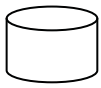
Decision (kondisi/perbandingan)



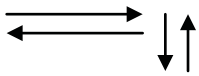
Preparation / Looping (perulangan)

Document

Connector (penghubung flow)

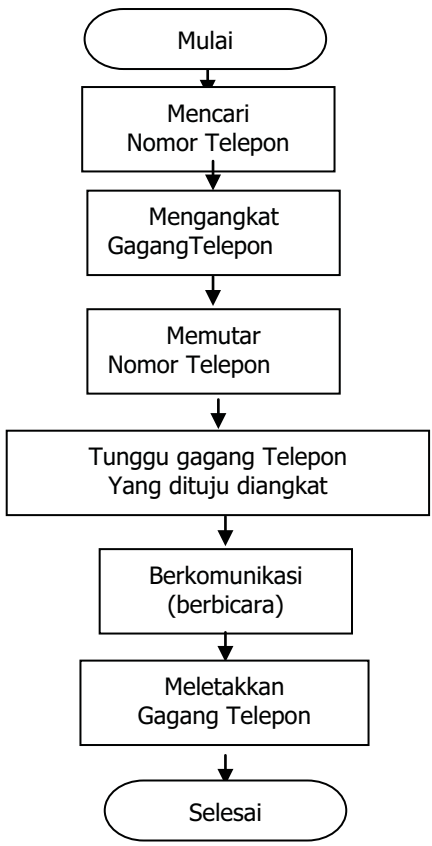


Magnetic Disk (database)



Data Flow (alir data)

Dapat digambarkan dalam bentuk diagram alur seperti berikut ini:



3. Dengan Pseudocode

Dengan menggunakan Pseudocode artinya algoritma tersebut mirip dengan statemen atau perintah pada bahasa pemrograman yang digunakan: contoh statement:

- INPUT (menerangkan data masukan)
- OUTPUT (data keluaran)
- PRINT (pencetakan ke input/output standar)
- FI (kondisi/seleksi)
- LOOP (perulangan/iterasi)

Contoh Kasus :

Penentuan Kelulusan Mata Kuliah berdasarkan Nilai Akhir

Kriteria : Jika NA >= 60 Maka Lulus

NA < 60 MAka Tidak Lulus

Permasalahan diatas dapat disajikan dengan 3 (tiga) cara algoritma yaitu:

1. *Aloritma Dengan Naratif*

Langkah 1 : Input Nilai TUGAS, UTS, dan UAS

Langkah 2 : Hitung Nilai Akhir dengan rumus

NA = (2 x TUGAS + 3 X UTS + 5 X UAS)/10

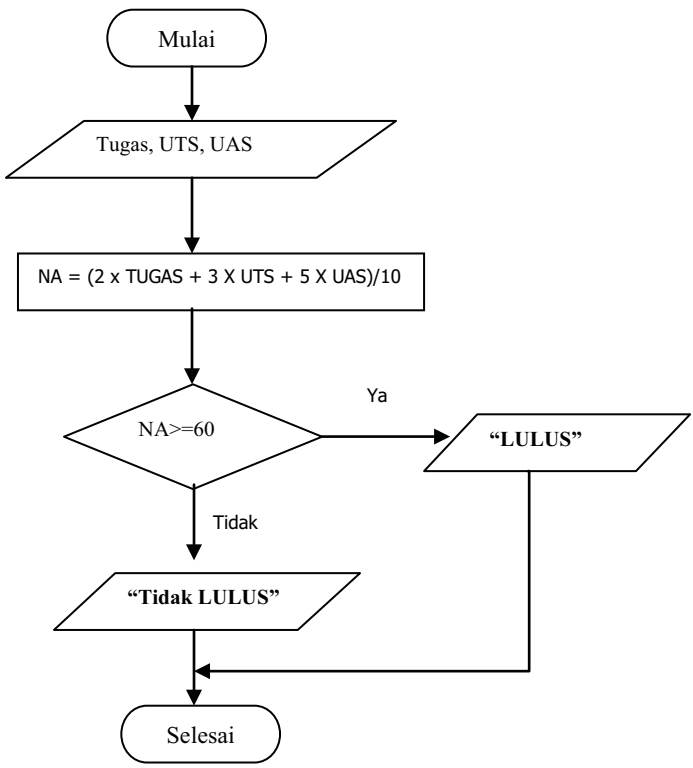
Langkah 3 : Menentukan kelulusan :

Jika NA >= 60 Maka "**LULUS**"

JIka NA < 60 Maka "**Tidak LULUS"**

Langkah 4 : Selesai

2. Aloritma Dengan Flowchart



3. Aloritma Dengan Pseudo Code

INPUT Tugas, UTS, UAS
Cout << "Nilai Tugas = "; Cin>>Tugas;
Cout << "Nilai U T S = "; Cin>>UTS;
Cout << "Nilai U A S = "; Cin>>UAS;
PROSES Nilai Akhir
NA = (2 x TUGAS + 3 X UTS + 5 X UAS)/10
UJI Kelulusan
IF (NA>=60)
Cout << " LULUS";
Else
Cout << " Tidak LULUS";

1.6 Tujuan Pemrograman

Seperti sudah dijelaskan pada pengertian pemrograman maka tujuan pemrograman itu sendiri adalah untuk mendapatkan hasil pekerjaan yang lebih efisien dan efektif dalam pengolahan data dengan menggunakan alat bantu komputer. Sehingga segala sesuatunya yang dikerjakan dengan komputer (*menggunakan program aplikasi*) jauh lebih mudah dalam menghasilkan output yang diinginkan jika dibandingkan dengan melakukan pengolahan data secara manual, sehingga dalam kegiatan pekerjaan yang jelas program sangat membantu user dalam mengolah data.

1.7 Manfaat Pemrograman

Manfaat pemrograman itu sangat berdampak positif, bahkan sangat membantu para pengelola data dalam menghasilkan output dari suatu pekerjaan. Seperti pengolahan data keuangan di perbankan, data kemahasiswaan di Perguruan Tinggi, kepegawaian di pemerintahan dan adminstrasi perkantoran, pengontrolan alat-alat industri, pengolahan data penjualan dan pengolahan data lainnya. Sehingga dengan adanya pemrograman ini pekerjaan sejumlah pegawai atau karyawan dapat dikerjakan oleh program tersebut dengan hasil yang lebih maksimal dan lebih akurat.

1.8 Tahapan Dalam Pemrograman

Tahapan yang dilakukan dalam penyelesaian masalah dengan komputer (dalam pemrograman) adalah:

1. *Pendefinisian Masalah*:
    - apa dan bagaimana masalahnya ?
    - apa masukan (*input*) yang diperlukan ?
    - apa Hasil yang diinginkan (*output*) ?

2. *Gambarkan Permasalahan Dalam Bentuk Diagram*
   - bagan penyelesaian secara keseluruhan
   - bagan detail dari masing-masing komponen yang terlibat dalam permasalahannya serta deskripsi tugasnya
3. *Pemilihan Metoda Penyelesaian*
   - struktur Data
   - langkah penyelesaian atau algoritma

4. *Pengkodean (Coding)*
   - memilih bahasa pemrograman yang sesuai
   - menerjemahkan algoritma ke dalam bahasa pemrograman tersebut

5. *Mencari Kesalahan*

   Cari Kesalahan-kesalahan
   - kompilasi: sintaks
   - pelaksanaan: simantik, logika/algoritma, ketelitian

6. *Uji dan Verifikasi*
   - Menjamin bahwa sampai ke tingkat kepuasan tertentu program dapat berjalan dengan baik dan benar,
   - dipastikan semua kemungkinan dapat memberikan hasil yang benar.

7. *Dokumen Tertulis*
   - pemakai: (manual petunjuk pemakaian program)
   - dokumentasi bagi pemrogram untuk keperluan perbaikan atau perubahan jika sewaktu-waktu diperlukan

8. *Pemeliharaan Program*
   - Memperbaiki kekurangan yang ditemukan kemudian
   - Modifikasi atas dasar perubahan dalam spesifikasi masalah.

BAB 2 BAHASA PEMROGRAMAN C++

2.1 Bahasa C++

Bahasa C++ merupakan salah satu bahasa pemrograman yang bersifat umum artinya dapat difungsikan untuk semua jenis permasalahan. Bahasa program ini termasuk bahasa program tingkat menengah (*medium level language*). Karena bahasa ini dapat mengakses atau menerjemahkan bahasa rakitan yang mendekati bahasa mesin dan juga mirip seperti bahasa tingkat tinggi dengan menggunakan bahasa manusia.

Kelebihan dari bahasa ini adalah hemat ekspresi, alur kontrol yang bagus dan struktur data modern serta kaya dengan operator. Bahasa ini memberikan kesempatan kepada pembuat program untuk lebih leluasa dalam memberikan instruksi dan dapat mengendalikan secara penuh. Penghematan ekspresi adalah hal yang disukai oleh para pembuat program (*programmer*) karena disamping singkat dan padat program dapat lebih mudah dipahami karena tidak membutuhkan baris program yang panjang.

Bahasa pemrograman ini berkembang dari Bahasa C++ itu sendiri sekaligus menyempurnakan kekurangan atau fasilitas yang tidak dimiliki oleh c khususnya dalam pemrograman berorientasi objek (*object oriented programming*).

2.2 Struktur Bahasa Program

Setiap bahasa pemrograman memiliki tiga struktur yaitu*, skuensial, pencabangan dan perulangan*. *Sekuensial* adalah konsep paling dasar dalam pemrograman dimana sekelompok perintah dilaksanakan komputer satu per satu perintah sebelum melaksanakan perintah berikutnya. Struktur *pencabangan* merupakan konsep yang lebih maju dan modern dibandingkan struktur sekuensial.

Cara ini dilakukan tidak harus semua program dibaca tetapi dengan melakukan seleksi atau pencabangan tertentu sehingga jika satu kondisi telah dipenuhi maka kondisi lain tidak diteruskan'dikerjakan lagi, prosesnya akan lebih cepat. Sedangkan struktur *pengulangan* adalah konsep yang lebih maju dari kedua struktur diatas yaitu program dapat melakukan pengulangan atau iterasi pada sebuah proses tertentu dan dapat mebedakan hasil proses setiap pengulangan. Jumlah atau batas pengulangan dapat ditentukan sesuai dengan jumlah yang diinginkan.

Sekarang ini pada setiap bahasa program harus menggunakan ketiga struktur ini agar dapat menjawab setiap persoalan yang muncul dalam pemrograman. Bahasa program juga akan dilengkapi dengan *pembuatan konstanta pendefenisian variabel, penggunaan operator, cara penulisan program atau sintaks, fungsi dan procedure.* Sehingga dalam mempelajari sebuah bahasa pemrograman langkah yang baik dilakukan adalah dengan megetahui ketiga struktur diatas serta dapat mengenal komponen yang tersedia dalam bahasa tersebut.

Struktur Program Bahasa C++

```
#include <headerfile1.h>          //deklarasi header file
#include <headerfile2.h>
.
.
#include <headerfileN.h>
#define c1 nilai_konstanta        //deklarasi konstanta
#define c2 nilai_konstanta
.
.
#define cN nilai_konstanta
typedef tipe_dasar tipe_baru      //deklarasi tipe
tipe variabel1 [=nilai_awal],     //deklarasi variabel
tipe variabel2 [=nilai_awal],
.
.
tipe variabelN [=nilai_awal];
tipe SubProgram1(parameter)       //deklarasi judul sub program
tipe SubProgram2(parameter)
```

```
.
.
tipe SubProgramN(parameter)
tipe main(parameter)              //Program Utama
{
//deklarasi variabel lokal
.
.
[return nilai;]
}
tipe SubProgram1(parameter)    //deklarasi isi sub program
{
//deklarasi variabel lokal
.
.
[return nilai;]
}
tipe SubProgram2(parameter)
{
//deklarasi variabel lokal
.
.
[return nilai;]
}
.
.
3
tipe SubProgramN(parameter)
{
//deklarasi variabel lokal
.
.
[return nilai;]
}
```

2.3 Sejarah Singkat C++

Bahasa C++ merupakan pengembangan dari bahasa pemrograman B yang disusun oleh *Ken Thomson* pada tahun 1970 untuk mesin DEC PDP-7 yang menggunakan sistem operasi Unix. Bahasa pemrograman B itu sendiri

diadaptasikan dari bahasa pemrograman BCPL (*Bahasa Combine Programming Language*) yang disusun oleh *Martin Richards* pada tahun 1967.

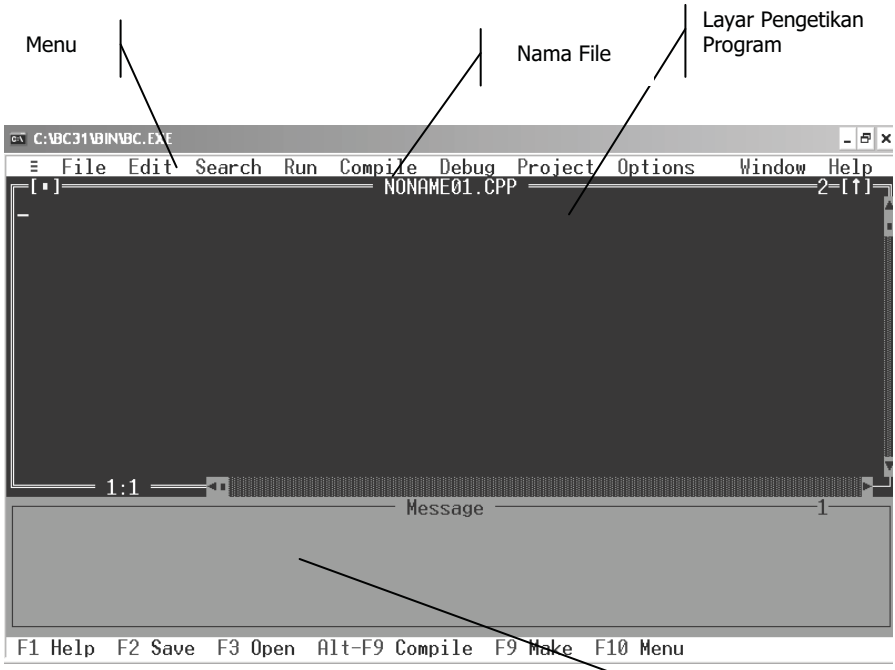
Kemudian C ini disusun oleh Dennis M Ritchi pada tahun 1972 . tidak lama setelah itu C diimplementasikan oleh Ritchie dalam menuliskan kembali sistem operasi UNIX yang sebelumnya ditulis dalam bahasa Asembler DEC PDP-11. Pada awalnya Bahasa C++ banyak dipakai untuk pemrograman sistem di laboratorium-laboratorium penelitian unversitas di Amerika Serika. Standar acuan yang digunakan adalah standar yang dikeluarkan bersama sistem operasi UNIX System 5 dan dijabarkan dalam buku *The C Programming Language oleh brian W. Kerninghan dan Dennis M, Ritchie*.

Sehingga dengan adanya C ini banyak para pengembang melakukan pengembangan melalui bahasa C++ untuk membuat perangkat lunak lainnya. Hingga tahun 80an berbagai versi C untuk lingkungan non UNIX untuk tingkat mikro komputer sudah banyak beredar dipasar perangkat lunak. Tidaklah salah kalau dikatakan bahwa versi-versi C untuk mikrokomputer inilah yang memegang peranan yang cukup besar dalam menunjang kepopulerannya.

Diantara Versi C untuk mikrokomputer yang umum dikenal adalah Turbo C dari Borlan International, Microsoft C dari Microsoft Corporation dan Microsoft QuickC, Lattice dari Lattice Incorporation dan Eco-C dari Ecosoft Incorporation.

Pada perkembangan selanjutnya C diarahkan ke pemrograman berorientasi objek (*Object Orientied Programming*). Beberapa versi mikrokomputer untuk pemrograman berorientasi objek adalah Turbo C++. Zortech C/C++, Borland C/C++ dan Microsoft C/C++. Dan Bahasa Pemrograman C++ dianggap sebagai superset dari C.

2.4 Tampilan Editor Program Bahasa C++



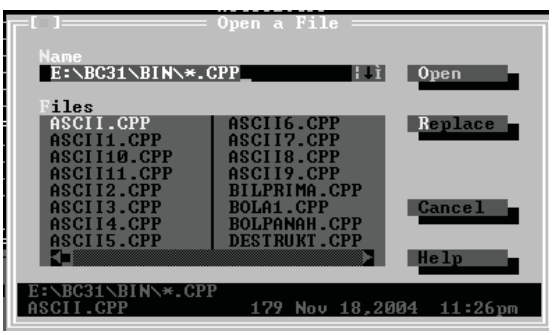
Gambar Layar Utama C++

Memulai file baru

- Menu File
- New

Membuka File

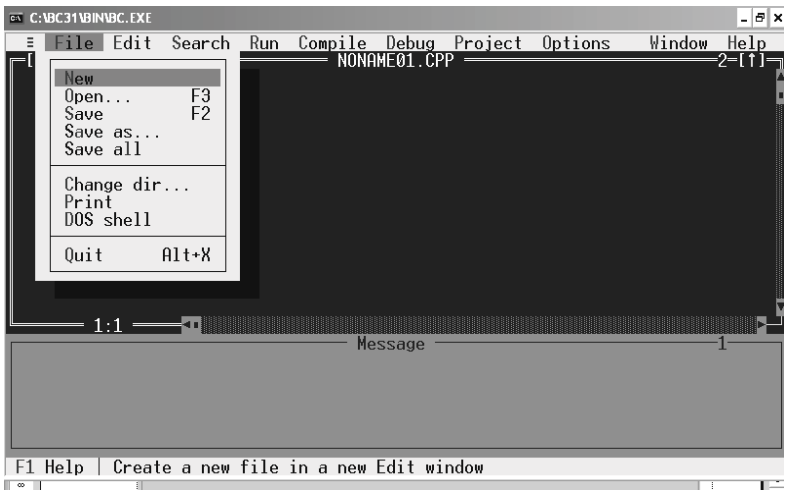
- Menu File
- Open, pilih File dengan ekstensi *.CPP



Gambar open file

2.5 Cara Penggunaan Menu

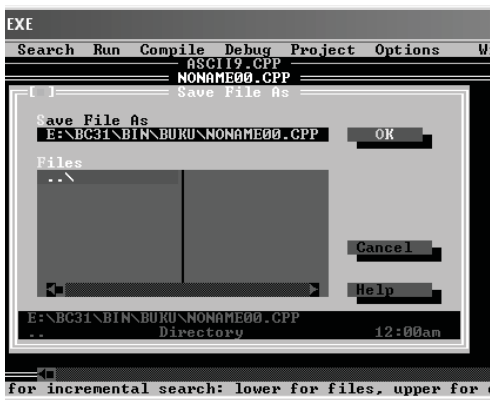
Penggunaan menu dapat diaktifkan dengan mengunakan mouse jika dijalankan dari Sistem Operasi Windows atau dengan menekan tombol Alt diikuti huruf awal menu seperti berikut ini:



Gambar mengaktifkan menu

Meyimpan file

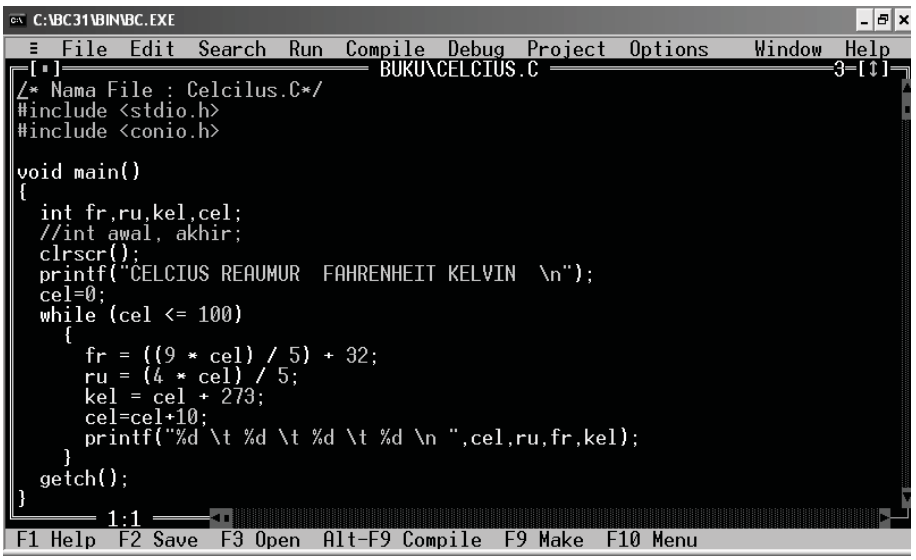
- Menu File
- Save
- Tentukan Folder dan Ketikkan nama file ekstensi *.CPP



Gambar menyimpan File

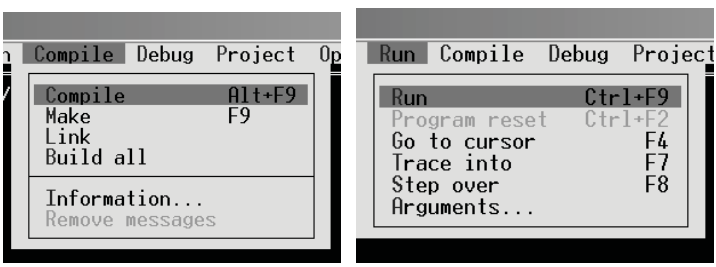
2.6 Contoh Penulisan Program

Penulisan program dapat dilakukan dengan pengetikan pada layar editor yang disediakan, biasanya pada tampilan standar fungsi dan kata kunci akan dibedakan warna dengan string atau teks lainnya. Sehinggga mudah mengetahui perbedaan fungsi-fugnsi tersebut.



Gambar Penulisan Program

Untuk melakukan eksekusi
Run → Ctrl + F9
Compile → Alt + F9
Lengkapnya pada tampilan berikut ini



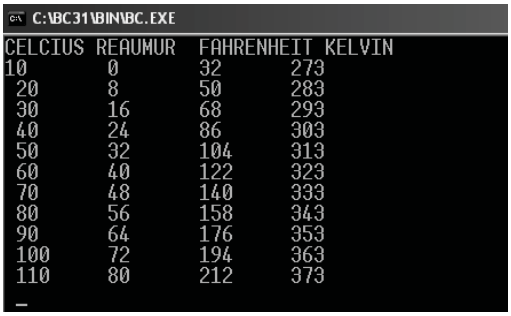
Gambar Menu compile program

Untuk memastikan kompiler dapat berjalan dengan baik maka seting file source dan file system pada menu Option, Directori sesuai dengan folder file Bahasa C++ di komputer yang digunakan.
Contoh

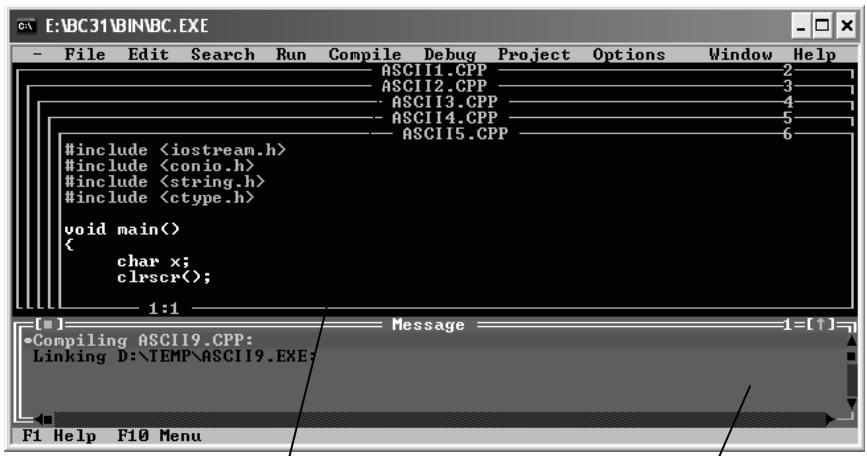
C:\TC\Include; untuk file include
C:\TC\LIB; untuk file library
D:\TEMP\Include; untuk file output file

Gambar pengaturan File compile, library dan output

Berikut ini contoh hasil eksekusi program pada layar komputer.



Gambar output Program



Gambar Pengaturan Layar

Multi Window

Pesan Kesalahan

Bahasa C++ memiliki dua buah komentar:

1. // Komentar ini hanya berlaku satu baris saja
   Contoh : count++; //count ditambahkan dengan 1
2. /* .. */ Komentar ini berlaku lebih dari satu baris
   Contoh :
   D=b*b-4*a*c
   /*bagian ini tidak dibaca oleh program*/

2.7 Keywoard Dalam C++

Selain dari pada ANSI C juga terdapat beberapa tambahan keywords yaitu:

asm	new	template
class	operator	this
delete	private	virtual
friend	protected	
inline	public	

BAB 3 ATURAN PENULISAN PROGRAM C++

3.1 Susunan Program C++

Setiap bahasa pemrograman memiliki aturan dan cara penulisan kode program yaitu pada tata urutan maupun sintaks. Bahasa C++ tersusun atas beberapa unsur penting yaitu:

- dokumentasi program,
- pengarah pengolahan,
- bagian deklarasi global,
- sebuah fungsi main(),
- fungsi buatan programmer

Setiap fungsi utama (*main*) maupun fungsi buatan terdiri dari badan fungsi yang memuat bagian deklarasi lokal dan kode program yang dapat dieksekusi. Secara umum susunan C++ dapat digambarkan seperi berikut ini:

```
[dokumentasi program]
[pengarah prapengolahan]
[deklarasi global]
[jenis-data] main[daftar argumen]
{
        [ deklarasi lokal]
        [kode program yang dapat dieksekusi]
}
[fungsi-fungsi buatan programmer]
```

keterangan: setiap kurung siku diatas bersifat opsional jka dibutuhkan maka dapat dituliskan dan jika tidak diperlukan maka tidak dituliskan.

3.2 Fungsi Utama

Fungsi utama merupakan fungsi induk dari sebuah program C++ yang harus ada pada setiap pembuatan Program. Bentuk penulisan adalah sebagai berikut:

```
void main()
{
        cout << "Selamat Belajar C++";
}
```

tanda () berfungsi untuk mengapit sebuah argumen fungsi yaitu sebuah nilai yang akan dileewatkan melalui fungsi.

3.3 Include File Header

#include <iostream.h> bukanlah sebuah pernyataan. Hal ini dapat ditunjukkan pada baris program yang tidak diakhiri dengan tanda titik koma (;). Baris ini berfugnsi untuk menyisipkan file lain pada saat kompilasi dilakukan. File yang memiliki ekstensi *.h disebut dengan file header. Yaitu fiel yang berisi tentang deklarasi, seperti fungsi, variabel dan sebagainya.
Bentuk umum adalah

```
#include <namafile.h>
```

contoh:

```
1. #include <iostream.h>
2. void main()
3. {
        a. clrscr();
        b. cout << " Selamat Menggunakan C++";
4. }
```

Berikut ini akan dibuat sebuah contoh lain yang dapat menggambarkan susunan dari setiap baris program.

Nama File : Linkaran.CPP

```
1   /* program untuk menghitung luas Lingkaran*/
2
3   #include <stdio.h>
4   #include <math.h>
5   #define  pi 3.14
6   #define kuadrat(x) ((0) * (x))
7
8   double luas;
9   double luas_lingkaran(double);
10
11  void main(int argc, char **argv)
12  {
13    double jarijari;
14
15    if (ardc<2)
16     {
17          cout << "Penggunaan : \n\t%s jarijari \n" <<  argv[0]);
18          exit(0);
19     }
20    jarijari=atof[argv[1]);
21     luas=luas_lingkaran(jarijari);
22     cout << "Luas Lignkaran berjari-jari %F  =" <<  jarijari,luas);
23  }
24
25  double luas_lingkaran(double r)
26  {
27    return (pi * kuadrat(r));
28  }
```

Pembahasan Linkaran.CPP

- /* program untuk menghitung luas Lingkaran*/
  dokumentasi program atau keterangan tentang tujuan program (komentar) tdk dieksekusi program
- #include <stdio.h>
- #include <math.h>
  Merupakan pengarah atau prapengolah dimana dalam file **#include** perintah atau sintaks penulisan perintah yang digunakan terdapat dalam file <stdio.h> dan <math.h> (*file source jika file ini tidak ada maka program tidak dapt dieksekusi atau tidak mengenal perintah yang dituliskan dalam listing program*)

- #define pi 3.14
  Defenisi konstanta artinya mendefenisikan sebuah variabel dengan nilai tertentu yang tetap

- #define kuadrat(x) ((0) * (x))
  Defenisi makro merupakan defenisi sebuah proses yang dapat digunakan pada program fungsi utama atau fungsi buatan.

- double luas;
- double luas_lingkaran(double);
  Bagian deklarasi global yang mendefenisiakn dua variabel yaitu luas bertipe double dan luas_lingkaran bertipe double yang dapat digunakan dalam setiap fungsi.

- void main(int argc, char **argv)
  fungsi main atau fungsi utama merukapan induk dari setiap fungsi buatan (fungsi utama dapat memanggil fungsi buatan)
  fungsi ini berjenis void dan menerima duabuah argumen
- {
  Blok program awal fungsi main

- double jarijari;
  bagian deklarasi lokal yang memuat deklarasi variabel jarijari bertipe double
- if (ardc<2)
  {
  cout << "Penggunaan : \n\t%s jarijari \n" << argv[0]);
  exit(0);
  }
  jarijari=atof[argv[1]);
  luas=luas_lingkaran(jarijari);
  cout << "Luas Lingkaran berjari-jari %F = %f \n" <<
  jarijari,luas;
  Bagian kode program merupakan proses yang dapat dieksekusi dan menghasilkan nilai, dalm kode ini terdapat fungsi if dan blok pernyataan dan memiliki struktur pencabangan.

- }
  Akhir blok program fungsi main

- double luas_lingkaran(double r)
  {
  return (pi * kuadrat(r));
  }
  fungsi buatan programmer, berjenis double dan dapat menerima argumen berjenis doubel dengan nama fungsi luas_lingkaran.

Contoh Program
Penggunaan printf

```
1.  #include<stdio.h>
2.  void main()
3.  {
4.    int A,B,C;
5.    A=6;
6.    B=4;
7.    C=A + B;
8.    printf("%i",C);
9.  }
```

Hasil tercetak : 10

Penggunaan sqrt()

```
1.  #include<stdio.h>
2.  #include<math.h>
3.  void main()
4.  {
5.    int A,B;
6.    A=25;
7.    B=sqrt(A);
8.    printf("%i",B);
9.  }
```

Hasil tercetak : 25

Penggunaan strcpy()

```
1.  #include<stdio.h>
2.  #include<string.h>
3.  void main()
4.  {
5.    Char A[5] = “ABCD”;
6.    Char B[5] = “PQRS”;
7.    Strcpy(B,A);
8.    puts (B);
9.  }
```

Hasil tercetak “ABCD”

BAB 4 JENIS DATA, VARIABEL & OPERATOR

Dalam melakukan pemrosesan data sebaiknya pembuat program komputer harus dapat membedakan satu jenis data dengan jenis data lainnya. Jika suatu program mengerjakan sebuah permasalahan dengan berbagai jenis data maka jika tidak sesuai dengan jenis datanya maka informasi yang dihasilkan pasti salah atau tidak sesuai dengan yang diinginkan. Dengan kata lain jika data masukan benar, proses benar maka data kelauran juga akan benar.

Dalam bahasa C++ untuk menangani jenis data maka sebaiknya menelusuri konsep *kata kunci, pengenal, jenis konstanta dan variabel, konstanta khusus, penamaan dan penggunaan Variabel, pendeklarasian variabel, pemberian nilai ke variabel dan konversi jenis data*.
Masing-masing konsep di atas akan dibahas berikut ini:

4.1 Kata Kunci

Kata kunci merupakan komponen pembentuk pernyataan (*state-ment*) yang memiliki arti khusus pada sebuah bahasa pemrograman. Berikut ini ada beberapa kata kunci (*keyword*) yang disepakati oleh ANSI C++ sebanyak 32 buah

auto	Const	double	float	Int	short	srtuct	unsigned
Break	Continue	else	for	long	signed	switch	void
Case	Default	enum	goto	register	sizeof	typedef	volatile
Char	Do	extern	if	return	statis	union	while

Semua kata kunci diatas harus dituliskan dalam bentuk huruf kecil

4.2 Pengenal

Berbeda dengan kata kunci yang telah memiliki arti dan kegunaan yang tetap, sebuah pengenal adalah kata yang digunakan oleh programmer untuk maksud dan tujuan tertentu. Sebuah pengenal merupakan kombinasi dari huruf, angka dan garis bawah penghubung dan huruf pertama harus diawali dengan huruf. Jumlah maksimum karakter yang digunakan untuk membentuk pengenal bervariasi tergantung dari kompiler yang digunakan. Standa ANSI memperkenalkan panjang maksimum 31 karakter.

Karakter yang digunakan dapat berupa huruf kecil atau huruf besar atau kombinasi keduanya dan tidak dapat menggunakan sapasi dengan kata lain harus satu kata. Tetapi dalam hal ini bahasa C membedakan antara huruf kecil dan huruf besar (*case sensitive*) contoh pemakaian pengenal yang mengacu ke nama variabel, konstanta bernama, nama fungsi dan nama larik (*array*) seperti contoh berikut ini:

- Indeks /* nama variabel */
- yN /* nama variabel */
- PI /* konstanta bernama */
- CariAngka() /* nama fungsi */
- JlhArr[10] /* nama array */

Pengenal tidak dapat menggunakan karakter khusus seperti
Sebagai contoh:

- luas lingkaran // salah
- jari-jari // salah
- luas_lingkaran // benar
- Nama1 // benar
- 1Nama // salah
- Angka# // salah
- Gaji$ // SAlah

4.3 Konstanta

Konstanta adalah sebuah sebuah objek yang memiliki nilai tertentu yang tidak dapat diubah. Sebuah konstanta buatan programmer yang harus memiliki nama, atau dapat berupa konstanta literal yang hanya merupakan representasi tertulis dari nilainya. Jenis data konstanta terdiri dari karakter, integer, floating point dan string.

Konstanta Karakter

Sebuah karakter diapit oleh tanda petik tunggal contoh,

'd','D','3','$'

karakter diatas merupakan karakter yang dapat dicetak, selain itu ada karakter yang tidak dapat dicetak seperti *escape, enter, bell, newline dan nul charakter* tipe yang seperti ini harus didahului dengan karakter backslash:

'\r','\a','\n', '\0'

Kelompok huruf

'a','b,'c'....'z'...'A','B'....''Z'

kelompok angka

'0','1','2'...'9'

kelompok karakter khusus

',' , '.' , '"" , '!', '@', '%'.....

Konstanta integer
Terdiri dari untaian bilangan bulat, berdasarkan basisnya konstanta integer biasa digunakan dalam Bahasa C dapat dibagi atas tiga

Integer Desimal (bilangan 0 – 9)
10. – 04, -1976, 10,564,358,-86

Integer Oktal (bilangan 0 – 7)
23,64,3,567,0,32

integer Heksadesimal (0 – F)
234.,456,AB23,43FA,A1,12,0..

a. Konstanta floating point
Adalah bilangan yang direpresentasikan dengan sebuah tanda positip dan negatip diikuti oleh bagian integer, titik desimal. Sebuah bilangan floating point dapat dinyakatan jika memiliki desimal.
1.3, 345.566 , 3.14563, dll

b. Deklarasi konstanta bernama

Pada bahasa C++ dapat dideklarasikan dengan dua cara yaitu:

1. Melalui pengarah #define
2. Melalui kata kunci Const

Contoh dengan menggunakan kata kunci #define

```
#define jlh_data        100
#define NamaFile "      Test.txt"
#define PI              3.14
```

dengan mengggunakan kata kunci Const

```
const int jlh_data = 100; double PI = 3.14;
void main(void)
{
}
```

4.4 Variabel

Variabel (*peubah*) adalah sebuah objek data yang dapat didefenisikan dan dinamai programmer secara eksplisit. Sebuah variabel menempati satu lokasi memori yang disisihkan untuk satu jenis data tertentu dan diberi nama untuk memudahkan pengacuannya. Nilai dari sebuah variabel dapat berubah jika terjadi pemasukan data juga berubah.

a. Dekaralasi variabel

Jenis_data nama_variabel
Contoh

```
Int nilai;
Double luas;
Char jk;
```

b. Klasisfikasi jenis variabel
Secara umum klasisifikasi jenis variabel terbagi atas tiga bagian:

1. Jenis fungsi (*dikenakan pada fungsi untuk mengetahui jenis data yang akan dikembalikan*)
2. Jenis Objek Data (*digunakan pada variabel yang digunakan dalam program*)
3. Jenis tidak lengkap (*digunakan jika struktur datanya belum lengkap seperti void*)

4.5 Tipe Data

Jenis	Deklarasi	Ukuran	Range
Karakter	(signed) Char	1 byte	-128 s/d 127
	Unsigned Char	1 byte	0 s/d 256
Integer	(signed) Int	2 byte	-32.768 s/d +32.767
	Unsgined Int	2 byte	0 s/d +65.535
Short	Short (signed)	2 byte	-32.768 s/d +32.767
	Unsgined Short	2 byte	0 s/d +65.535
Long	(signed) Long	2 byte	-2.147.483.648 s/d +2.147.483.648
	Unsgined Long	4 byte	0 s/d 4.294.967.295
Floating Point	Float	4 byte	3,4E-38 s/d 3,4E+38
	Double	8 byte	1,7E-308 s/d 1,7E+308
	Long double	10 byte	3,4E-4932 s/d +1,1E+4932 (19 angka ketelitian)

Syarat penyusunan pengenal (*variabel & konstanta*)

1. Hanya boleh terdiri dari huruf, angka dan garis bawah dan tidak boleh diawali oleh angka
2. Tidak boleh menggunakan karakter khusus seperti #,$,@, dll
3. Dua pengenal yang dibedakan dengan huruf besar dan kecil adalah dua pengenal yang berbeda (case sensitive)

A. Operator Aritmatik

No	Operator	Contoh	Arti/Fungsi
1	+	c=a+b;	Operator Penjumlahan
2	-	c=a-b;	Operator Pengurangan
3	*	c=a*b;	Operator Perkalian
4	/	c=a/b;	Operator Pembagian
5	%	c=5%2;	Modulus (Mengambil sisa bagi) dari 5/2
6	=	c=10;	Operator Assignment
7	++	i++;	i=i+1;
8	--	i--;	i=i-1;
9	+=	c+=3;	c=c+3;
10	-=	c-=2;	c=c-2;
11	*=	c*=3;	c=c*3;
12	/=	c=/2;	c=c/2;
13	%=	c%=4;	c=c%4;

B. Operator Logika

No	Operator	Contoh	Arti/Fungsi
1	==	if (a==b)	Jika a sama dengan b
2	!=	if (a!=b)	Jika a tidak sama dengan b
3	<	c=a<b;	Jika a lebih kecil dari b, maka c bernilai 1, jika sebaliknya maka c bernilai 0
4	>	while (c>10) c--;	Kurangi c dengan 1 jika c lebih besar dari 10
5	<=	c=a<=b;	Lebih kecil atau sama dengan
6	>=	c=a>=b;	Lebih besar atau sama dengan
7	&&	if (n>80 && n<=100)	Jika n lebih besar dari 80 **dan** n lebih besar atau sama dengan 100
8 10	||	while (i	Tambahkan i selama i belum mencapai p **atau** ch bukan tombol ESC
11	!	if (!(a>='0' && a<='9'))	Jika a tidak ada diantara '0' dan '9'

C. Operator Logika

No	Operator	Contoh	Dalam Biner	Arti/Fungsi
1	&	a=6&9;	0110 & 1001 = 0000	a menghasilkan 0
2	|	a=6|9;	0110 | 1001 = 1111	a menghasilkan 15
3	!	b=!6;	!0110 = 1001	b menghasilkan 9
4	^	b=9^3;	1001 ^ 0011 = 1000	b menghasilkan 8
5	<<	c=3 << 2;	0011 digeser 2 bit ke kiri = 1100	c menghasilkan 12
6	>>	c=12 >> 2;	1100 digeser 2 bit ke kanan = 0011	c menghasilkan 3

Contoh Program untuk mencetak ASCII Lengkap

```
1   #include <iostream.h>
2   #include <conio.h>
3
4   void main()
5   {
6
7     unsigned char ch;
8     clrscr();
9     cprintf("Tabel ASCII \n");
10    cout << "\n";
11
12    for (ch=0; ch<255; ch++)
```

```
13              {
14               // jika karakter line feed
15               if (ch==0)
16               {
17               //  cprintf("<LF>");
18                 cout << "=");
19                 cout <<  "%-3d " <<  ch;
20               }


21              // jika karakter enter
22               if (ch==13)
23               {
24                 cout <<  "<LF>";
25                 cout << "=";
26                 cout <<  "%-3d " << ch);
27               }
28               cprintf("%c",ch);
29               cprintf("=");
30               cprintf("%3d ", ch);
31
32              if ((( ch % 12 ) == 0) && (ch > 0))
33              cout <<  "\n";
34
35              }
36             getch();
37           }


1          /* Nama File : Celcilus.CPP*/
2          #include <constrea.h>
4          //gabungan iostream.h dan conio.h
5          void main()
6          {
7            int fr,ru,kel,cel;
8            //int awal, akhir;
9            clrscr();
10           cout<< "CELCIUS REAUMUR  FAHRENHEIT KELVIN  \n" ;
11           cel=0;
12           while (cel <= 100)
```

```
13          {
14           fr = ((9 * cel) / 5) + 32;
15           ru = (4 * cel) / 5;
16           kel = cel + 273;
17           cel=cel+10;
18           cprintf("%d \t %d \t %d \t %d \n ",cel,ru,fr,kel);
19          }
20         getch();
21        }
```

```
1       /* Nama File : Bulan.CPP*/
2       /* dengan penggunaan fungsi enum*/
3
4       #include <iostream.h>
5       #include <conio.h>
6         enum bln ubahBulan(enum bln);
7
8       void main()
9       {
10        /*Prototipe Fungsi*/
11        // defenisi jenis data enum
12        enum bln {jan,feb,mar,apr,mei,jun,jul,agt,sep,okt,nov,des};
13        //deklarasi variabel berjenis enum
14        enum bln bln_sekarang,bln_lalu;
15        bln_sekarang = apr;
16        bln_lalu =ubahBulan(bln_sekarang);
17        cprintf("sebelum bulan ke %-d adalah
18              bulan ke %-d \n",bln_sekarang+1, bln_lalu + 1);
19        getch();
20      }
21        /* fungsi untuk menghitung bulan sebelumnya */
22        enum bln ubahBulan(enum bln bln1)
23        {
24         bln1--;
25         return bln1;
26        }
```

```
1   /* Nama File : Htkarakt.CPP */
2   /* untuk menghitung jumlah karakter*/
3
4   #include <iostream.h>
5   #include <conio.h>
6
7   void main()
8   {
9     char ch;
10    int jlbrs=1,jlSpasi=0,jllain=0;
11    clrscr();
12
13    cout << "Tuliskan sembarang kalimat  :\n";
14    cout << "Berikan *, Jika selesai \n";
15
16    while ((ch = getchar()) != '*')
17      {
18       if (ch== ' ')
19         jlSpasi++;
20       else if (ch == '\n')
21
22       else
23
24      }
25    cprintf("\jumlah spasi        = %d \n ", jlSpasi);
26    cprintf("\jumlah baris        = %d \n ", jlbrs);
27    cprintf("\jumlah karakter lain = %d \n ", jllain);
28
29    getch();
30  }
```

BAB
5 OPERATOR DAN EKSPRESI

Setiap bahasa pemrograman memiliki penghubung/operator untuk dapat mengolah data sesuai dengan bentuk keluaran yang didinginkan. Operator adalah simbol-simbol yang digunakan untuk mengolah data tersebut. Bahasa C++ memiliki banyak jenis operator dibandingkan dengan bahasa program lainnya. Jenis operator tersebut berupa :

- operator aritmatika,
- operator relasional
- operator logika
- operator khusus
  - operator sizeof
  - operator penugasan
  - operator koma
  - operator bertingkat
  - operator alamat
  - operator bersyarat

5.1 Operator Aritmatika

Operator ini digunakan untuk mengolah data aritmatis.

Operator	Nama	Contoh	Penjelasan
*	Perkalian	I * j	Kalikan i dengan j
/	Pembagian	I / j	Bagikan i dengan j
%	Modulo	I % j	Sisa pembagian i dengan j
+	Penjumlahan	I + j	Tambahkan i dengan j
-	Pengurangan	I – j	Kurangkan i dengan j
++	Postfix inkremen	I++	Inkremen i setelah digunakan
	Prefix inkremen	++i	Inkremen i sebelum digunakan
--	Postfix Dekremen	i--	Dekremen i setelah digunakan
	Prefix Dekremen	--i	Dekremen i sebelum digunakan
-	Negasi	-i	Negasikan nilai i
+	Unary Plus	+i	Nilai positif dari i

5.2 Operator Relasional

Operator ini digunakan untuk mengolah data yang menggunakan pembadingan suatu nilai.

Operator	Nama	Contoh	Penjelasan
>	Lebih besar	I > j	Jika i lebih besar dari j
>=	Lebih besar atau sama dengan	I >= j	Jika i lebih besar atau sama dengan j
<	Lebih kecil	I < j	Jika i lebih kecil dari j
<=	Lebioh kecil atau sama dengan	I <= j	Jika i lebih kecil atau sama dengan j
==	Sama dengan	I == j	Jika i sama dengan j
!=	Tidak sama dengan	I != j	Jika i tidak sama dengan j

5.3 Operator Logika

Operator ini digunakan untuk mengolah dengan operasi logika.

perator	Nama	Contoh	Penjelasan
!	Not Logika	!i	Jika tidak i
&&	And Logika	i && j	Jika i dan j
||	Or Logika	i || j	Jika i atau j

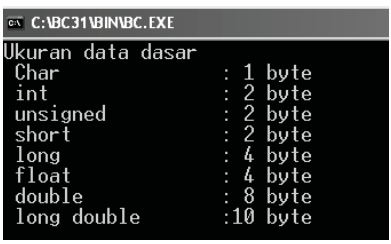
5.4 Operator Sizeof

Operator ini digunakan untuk perhitungan jumlah byte yang dibutuhkan menyimpan sebuah objek atau jenis data. Cara kerja dan bentuk pemanggilan operator size of mirip seperti kerja dan bentuk pemanggilan fungsi standar, yaitu:

Sizeof(objek data) atau sizeof (jenis data)

```
1.  /* untuk menghitung ukuran byte jenis data*/
2.  #include<stdio.h>
3.  //pengaturan control standart input/output
4.  #include<conio.h>
5.  void main(void)
6.   {
7.    // untuk membbersihkan tampilan diawal terlebih dahulu
8.    clrscr();
9.    cprintf ("Ukuran data dasar \n");;
10.   cprintf (" Char      : %2d byte \n", sizeof(char));
11.   cprintf (" int       :%2d byte \n", sizeof(int));
12.   cprintf (" unsigned  :%2d byte \n", sizeof(unsigned));
13.   cprintf (" short     :%2d byte \n", sizeof(short));
14.   cprintf (" long      :%2d byte \n", sizeof(long));
15.   cprintf (" float     :%2d byte \n", sizeof(float));
16.   cprintf (" double    :%2d byte \n", sizeof(double));
17.   cprintf (" long double:%2d byte \n", sizeof(long double));
18.   // untuk menerima masukan sembarang karakter dan menahan
19.   // tampilan dilayar
20.   getch();
21.  }
```

Output :



5.5 Operator Penugasan

Operator ini digunakan untuk memberikan nilai pada suatu variabel. Operator ini dengan sisi sebelah kirinya berupa lvalue. Yaitu sebuah variabel tunggal yang mengacu ke memori komputer, sementara sisi sebelah kanan dapat berupa lvalue atau rvalue. Rvalue adalah sebuah ekspresi yang bukan lvalue.

Contoh pemakaian operator penugasan:

```
I=0;
Tambah = ++hitung;
Total = total + 1;
LuasSegitiga = ( LuasAlas * Tinggi ) * 0.5;

5 = x;        // salah nilai tidak boleh dikiri
i * j =10;    // salah proses tidak boleh dikiri
```

Operator	Contoh	Ekspresi ekivalen	Penjelasan
+=	i+=n	i = i + n	Tambah i dengan n
-=	i-=n	i = i -n	Kurangi i dengan n
*=	i*=n	i = i * n	Kalikan ij dengan n
/=	i/=n	i = i / n	Bagikan i dengan n
%=	i%=n	i = i % n	Pembagian sisa i dengan n

5.6 Operator Koma

Operator ini digunakan untuk:

- memisahkan sederatan nama variabel dalam sebuah deklarasi
- memisahkan argumen fungsi
- menyatukan dua ekspresi menjadi sebuah pernyataan
- memungkinkan pemberian lebih dari satu ekspresi

contoh operator koma,

```
int bil,total;                          //pemisah variabel
printf("Jumlah Bilangan : %d \n",bil)   //pemisah argumen fungsi
```

5.7 Operator Bertingkat

Operator ini digunakan pada beberapa operator yang dikenakan sebuah ekspresi. Masing-masing operator dalam ekspresi tersebut memberikan nilai yang kemudian dapat digunakan di dalam ekspresi yang lebih besar. Sebagai contoh pada struktur perulangan.

while((c=getchar()) != '*')

Operator pada penugasan c = getchar() diatas berada dalah ekspresi lain yang memuat operator relasional tidak sama dengan. Urutan kerjanya:

1. memanggil fungsi standar c =getchar() dan memberi nilai karakter
2. menyimpan hasilnya dalam c dan
3. membandingkan c dengan '*' menghasikan nilai benar atau salah

5.8 Operator Alamat

Operator alamat adalah operator yang memberikan alamat dari operandnya, misalnya **&**i memberikan alamat i. Operand yang dikenal pada **&** merupakan lvalue yang memiliki tempat penyimpanan dalam memori.

5.9 Operator Bersyarat

Operator bersyarat merupakan satu-satunya operator yang bersifat triadic (*ternary operator*) yang membutuhkan tiga buah operand dan dikenal juga sebagai ekspresi bersyarat. Ketiga buah operand yang dibutuhkan pada operator ini terdiri dari satu ekspresi yang akan diuji dan dua ekpresi pilihan
Bentuk umum:

Kondisi ? expr1 : expr2

```
1   /* untuk membandingkan 3 buah bilangan paling besar*/
2
3   #include<constrea.h>
4
5   void main(void) //void untuk tipe data yagn tidak jelas
6    {
7     int a,b,c;
8
9     cout << "Pembandingan Bilangan \n";
10    cout << " \nMasukkan Bilangan A : ";
11    cin >> a;
12    cout << " \nMasukkan Bilangan B : ";
13    cin >> b;
14    cout << " \nMasukkan Bilangan C : ";
15    Cin >> c;
16    // proses pembadingan  dengan menggunakan operator AND
17    if (a > b && a > c)
18       {
19         cout << " \n Bilangan A Paling Besar ";
20       }
21    else if (b>a && b>c)
22       {
23         cout << " \nBilangan B paling Besar";
24       }
25    Else
26       {
27         cout << "\nBilangan C paling Besar";
28       }
29  }
```

5.10 Ekspresi

Ekspresi adalah gabungan dari antara operator dan operand. Dari penggabungan operator dan operan ini dapat dibagnun ekspresi-ekspresi yang lebih rumit. Namun bisa saja dalam bentuk yang paling sederhana.

Sebagai contoh,

```
A= b + 4;
X = --k % 2;
(x>= 0) || ((x %11) == 0);
/* Contoh Program penggunaan Operator Aritmatika */
```

```
1   #include<constrea.h>
2
3   void main(void)
4   {
5     int x,y=10,z=20;
6     x=z++;
7     y=y%4;
8     cprintf("x= %d, y= %d, z= d \n", x, y, z);
9     X=++z;
10    Y=x/2;
11    cprintf("x= %d, y= %d, z= d \n", x, y, z);
12    X=--y;
13    Z = z * 2;
14    cprintf("x= %d, y= %d, z= d \n", x, y, z);
15    X=y--;
16    Z= y + 1;
17    cprintf("x= %d, y= %d, z= d \n", x, y, z);
18    X=++z;
19    getch();
20  }
```

BAB 6 FUNGSI INPUT DAN OUTPUT

Hasil pengolahan data pada komputer harus ditampilkan pada layar serta dapat menerima data masukan dari pengguna program untuk diolah. Sehingga proses pemasukan data kemudian diproses/diolah lalu ditampilkan hasilnya. Untuk dapat melakukan ini fungsi dalam C ada beberapa fungsi seperti printf(), putchar(), putch() yang berfungsi untuk menampilkan kelayar. Fungsi untuk input atau masukan dari user seperti scanf(), getchar(), getch(), gets(). Berikut ini akan dibahas tentang fungsi diatas dan contohnya:

6.1 Fungsi Output

a. Fungsi cprintf()
   Fungsi ini paling banyak digunakan karena memiliki format sehingga disesuaikan dengan apa bentuk output yang diinginkan.
   Contoh
   cprintf("\n")
   cprintf ("Good Luck For You");
   cprintf("%.4.1f = %4.1f / %4.1f ",x,y,z);
   cprintf("Hasil Jumlah = %d \n",jlh);

   Argumen yang diberikan pada fungsi printf adalah tanda kurung. Serta jika ada komentar berupa string diapit tanda ("string..... ")

Berikut ini tabel lengkap penentu format yang dapat digunakan pada fungsi cprintf()

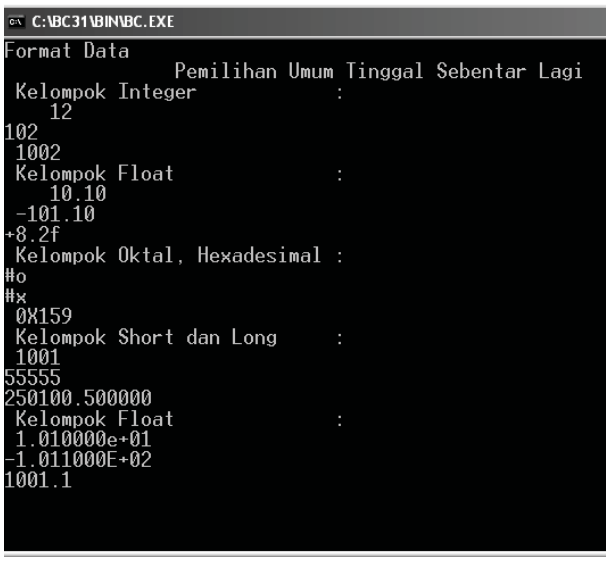
Penentu Format	Jenis Data	Spesifikasi Konversi
%c	Char	Karakter tunggal
%s	Char[]	String berupa konstanta
%d atau %i	Int	Integer desimal
%f	Float/double	Floating point
%e atau %E	Float/double	Notasi ilmiah
%n	Pointer ke int	Menyimpan jlh char dicetak
%g atau %G	Float/double	Floating point
%u	Unsigned	Integer desimal
%x atau %X	Unsigned	Untuk heksadesimal huruf keci/besar
%o	Unsigned	Integer oktal
p	Far pointer ke void	Alamat dicetak bentuk segment/offset

Contoh Program

```
1   /* untuk penggunaan penentu format*/
2
3   #include<constrea.h>
4   //pengaturan control standart input/output
5   #include<conio.h>
6
7   void main(void)
8    {
9     char str[] = "Pemilihan Umum Tinggal Sebentar Lagi";
10    int i=12, j=102, k =1002;
11    float x=10.1, y = -101.1, z=1001.1;
12    unsigned l =345, m = 0xf01c, n=0xafde;
13    short p =1001;
14    long q = 55555;
15    long double r = 250100.5;
16
17    // untuk membersihkan tampilan diawal terlebih dahulu
18    clrscr();
19
20    cprintf ("Format Data \n");;
21    cprintf (" %50s \n", str);
22    cprintf (" Kelompok Integer   : \n %5i  \n%-5d  \n%5d  \n", i,j,k);
```

```
23      cprintf (" Kelompok Float     : \n %8.2f\n%+8.2f\n+8.2f\n", x,y,z);
24      cprintf (" Kelompok Oktal, Hexa: \n#o    \n#x    \n %#X \n", l,m,n);
25      cprintf (" Kelompok Short/ Long: \n %hd  \n%ld   \n%Lf  \n", p,q,r);
26      cprintf (" Kelompok Float     : \n %e   \n%#E   \n%HG  \n", x,y,z);
27
28      // untuk menerima masukan sembarang karakter dan menahan
29      // tampilan dilayar
30      getch();
31    }
```

Output:



Selain dari konversi fungsi cprintf() juga dapat menerima karakter khusus disebut dengan karakter *escape* atau *escape sequence* berguna untuk menampilkan karakter yang tidak dicetak secara langsung. Karakter ini harus didahului tanda backslash "\"

Berikut ini daftar lengkap escape sequence :

Escape Sequence	Karakter
\a	Bel
\b	Backspace
\f	Formfeed
\n	Baris baru
\r	Carriage return
\t	Horizontal tab
\v	Vertical tab
\\	Backslash (\n)
\?	Tanda tanya
\'	Tanda kutip tunggal `
\"	Tanda kutip ganda
\0oo	Bilangan oktal
\xhh	Bilangan hexadesimal

b. Fungsi puts()

Fungsi puts adalah fungsi yang dapat dgigunakan untuk menuliskan string ke layar dan menambahkan baris baru. Contoh,

```
Puts("Tekan spasi untuk keluar..");
```

c. Fungsi cout <<

Fungsi ini paling banyak digunakan karena penggunaan sangan mudah dan merupakan ciri dari c++ itu sendiri.

Contoh

```
cout << "fungsi keluaran pada layar")
cout <<  "Good Luck For You");
cout << " Belajar C++" << "Sangat Asik"
cout << endl //untuk menurunkan pada baris berikutnya (end line)
```

6.2 Fungsi Input

a. Fungsi cin>>

Fungsi ini merupakan pasangan dari fungsi cout yang bertujuan untuk menerima masukkan dari alat input. fungsi ini sangat baik digunakan untuk pengolahan data numerik.bentuk penulisannya adalah sebagai berikut:

```
cin >> nama_Variabel;
contoh
cin >> a;
cin >> jumlah;

cout << "Masukkan Bilangan 1 : "
cin >> bil1;
cout << "Masukkan Bilangan 2 : "
cin >> bil2;
hasil = bil1 + bil2;
cout << "Hasil penjumlahan bil1 dan bil2 = " << hasil
```

b. Fungsi Scanf()

Fungsi ini paling banyak digunakan karena memiliki banyak fasilitas konversi ke dalam nilai yang diinginkan dengan format tertentu.

Fungsi scanf() memiliki dua ciri yaitu daftar argumen sepanjang apapun dapat di scan dan keua input yang diberikan dikendalikan oleh sebuah spesifikasi konversi atau format yang sederhana.

```
Contoh
Char ch1,ch2,ch3,str[100];
Int x;
Double z;
```

Dengan menggunakan fungsi scanf() nilai untuk variabel tersebut diterima sekaligus
Scanf("%c %c %c %s %d %4f", &ch1, &ch2, &ch3, &str, &x, &z);
Dari perintah diatas string kendali adalah "%c %c %c %s %d %4f" sementara argumen lain adalah &ch1, &ch2, &ch3, &str, &x, &z

String kendali tersusun atas tiga jenis pengarah, karakter biasa, whitespace dan sepesifikasi konversi. Sebagai contoh,

Scanf("Rp. %f", &Gaji);
Scanf("%d, %d", &x, &y);

Berikut ini tabel lengkap penentu format yang dapat digunakan pada fungsi scanf()

Penentu Format	Jenis Data	Spesifikasi Konversi
%c	Char	Karakter tunggal
%s	Char[]	String berupa konstanta
%d	Int	Integer desimal
%f	Float/double	Floating point
%e atau %E	Float/double	Notasi ilmiah
%n	Pointer ke int	Menyimpan jlh char dicetak
%g atau %G	Float/double	Floating point
%u	Unsigned	Integer desimal
%x atau %X	Unsigned	Untuk heksadesimal huruf keci/besar
%o	Unsigned	Integer oktal
p	Far pointer ke void	Alamat dicetak bentuk segment/offset

Contoh program:

```
1   /* Nama File : ctScanf.C*/
2   #include <stdio.h>
3   #include <conio.h>
4
5   void main()
6   {
7     int fr,ru,kel,cel;
8     //int awal, akhir;
9     clrscr();
10    cprintf("Masukkan Derajat Celcius   :");
11    scanf("%d",&cel);
12
13       fr = ((9 * cel) / 5) + 32;
14       ru = (4 * cel) / 5;
15       kel = cel + 273;
16       cel=cel+10;
17
18       cprintf(" Celcius    = %d\n ",cel);
19       cprintf("Reaumur    = %d\n ",ru);
20       cprintf("Fahrenheit = %d\n ",fr);
21       cprintf("Kelvin     = %d\n ",kel);
22
23    getch();
24  }
```

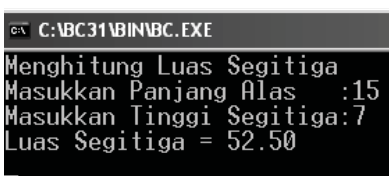
Output Program:

```
C:\BC31\BIN\BC.EXE
Masukkan Derajat Celcius   :100
 Celcius    = 110
 Reaumur    = 80
 Fahrenheit = 212
 Kelvin     = 373
```

Contoh 2:

```
1       //Program untuk menghitung luas segitiga
2       #include <constrea.h>
3       #include <conio.h>
4
5       void main()
6       {
7        float luas,alas,tinggi;
8        clrscr();
9        cprintf("Menghitung Luas Segitiga \n");
10       cprintf("Masukkan Panjang Alas   :");
11       scanf("%f",&alas);
12       cprintf("Masukkan Tinggi Segitiga:");
13       scanf("%f",&tinggi);
14       luas = (alas * tinggi) / 2.0;
15       cprintf("Luas Segitiga = %2.2f \n",luas);
16       getch();
17      }
```

Output program




```
1.  // Program untuk mengatur Warna
2.
3.  #include <constrea.h>
4.  #include <graphics.h>
5.  void main()
6.  {
7.      int i;
8.      clrscr();
9.      cout << "Masukkan Pilihan [1,2,3]  : ";
10.     cin >> i;
11.      if (i==1)
```

```
12.       {
13.          textcolor(2);
14.         textbackground(7);
15.         cprintf("salemat");
16.         cout << "\n";
17.       }
18.  else
19.  if (i==2)
20.    {
21.       textcolor(5);
22.      textbackground(2);
23.      cprintf("selamat");
24.      cout << "\n";
25.   }
26.  else
27.  if (i==3)
28.  {
29.     textcolor(4);
30.     textbackground(6);
31.     cprintf("selamit");
32.     cout << "\n";
33.  }
34.  else
35.     cprintf("\n\nTdak ada Pilihan");
36.  getch();
37.  }
```

BAB 7 SELEKSI DAN PECABANGAN

Dalam setiap bahasa pemrogaman menggunakan cara untuk melakukan pengambilan keputusan dalam memilih salah satu syarat. Seleksi sering juga disebut sebagai kondisi. Berikut ini akan dibahas mengenai kondisi dalam dua bagian yaitu dengan mengunakan perintah **if, if...else. Else-if** dan perintah **switch, break,continue dan goto.** Dalam kehidupan sehari-hari kondisi juga sering kita gunakan dalam melakukan pilihan terhadap suatu makanan, minuman, pakaian dan sebagainya.

Kondisi biasanya selalu diuji dengan hal yang tertentu sesuai dengan syarat yang diberikan. Contoh sebuah peryataan berkondisi "Kalau ada buku Bahasa C maka saya akan mempelajarinya tetapi kalau buku Bahasa C tidak ada maka saya akan mempelajari buku yang lain". Apabila pernyataan ini diterjemahkan kedalam algoritma maka akan kita peroleh seperti berikut ini:

```
Jika Buku  Bahasa C++ = "Ada" Maka
   Proses = "Saya akan mempelajarinya"
Jika tidak
   Proses = "Mempelajari buku yang lain"
```

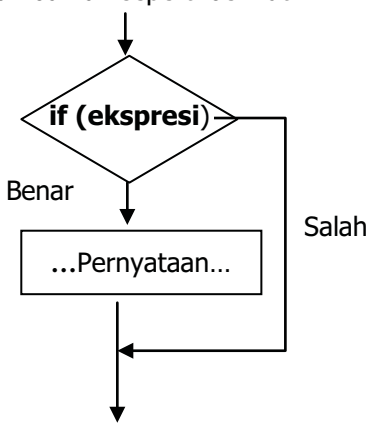
7.1 Struktur if

Statemen if digunakan untuk membandingkan suatu kondisi ke kondisi tertentu. Jika kondisi tersebut sesuai dengan pembandingnya maka akan dijalankan perintah tertentu. Tetapi apabila tidak memenuhi syarat atau tidak sesuai dengan pembandingnya maka kondisi tersebut salah dan akan dilanjutkan ke kondisi yang berikutnya serta perintah yang diberikan tidak dijalankan.

Sintaks pernyataan if :

```
if (ekspresi)
{
   .. Peryataan1…
   .. Peryataan2…
}
```

Kondisi ini dapat digambarkan seperti berikut ini:



Gambar Alur eksekusi statemen if

Sebagai contoh: Menentukan bilangan ganjil atau genap:

```
if  (Bilangan % 2 == 0)
 {
   printf("Bilangan Genap");
 }
```

Pengertian : jika sebuah bilangan memiliki sisa pembagian 0 dengan 2 maka bilangan itu adalah bilangan genap dan sebaliknya jika sisa pembagian bilangan dengan 0 sama dengan 1 maka tidak ditampilkan.

Contoh lain

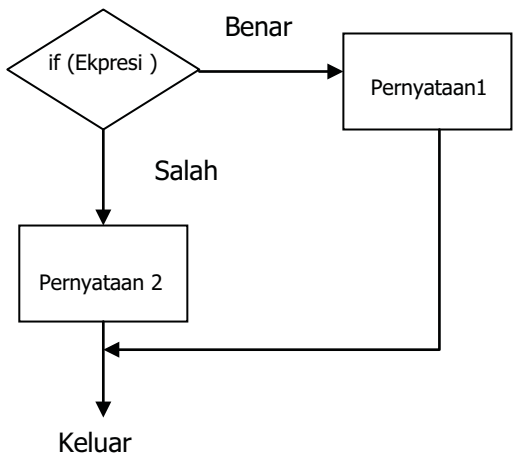
```
if scanf("%d", &n) !=1)
   cPrintf("Jenis input yang diberikan salah \n");
```

7.2 Struktur if-else

Kondisi if-else sebenarnya sama dengan kondisi if tunggal, hanya dibedakan dengan jumlah pilihan kondisi pada if else lebih dari satu sehingga jika pilihan pertama tidak dipenuhi akan dilanjutkan ke pilihan berikutnya:

Sintak pernyataan if else adalah sebagai berikut:

```
if    (ekspresi1)  Then
  {
    …pernyataan1…
  }
else
   {
     .. pernyataan2…
   }
```



Gambar Alur eksekusi if else

Contoh Progam:

```
if  (Bilangan % 2 == 0)
  {
    printf("Bilangan Genap");
  }
else
  {
    printf("Bilangan Ganjil");
  }
```

7.3 Struktur else - if

Jika sebuah pernyataan else-if mempunyai struktur if lain di dalam pernyataan else-nya. Pilihan ini dilakukan jika syarat yang akan dipenuhi lebih dari dua pernyataan.

Sintak pernyataan if else-if adalah sebagai berikut:

```
if    (ekspresi1)  Then
   {
      …pernyataan1…
     … pernyataan2…
   }
else if  (ekspresi2)
    {
      .. pernyataan1…
      .. pernyataan1…
    }
else if  (ekspresi3)
    {
      .. pernyataan1…
      .. pernyataan1…
    }
else
    {
      .. pernyataan1…
      .. pernyataan1…
    }
```

Contoh program

Buatlah satu program penilaian pada dengan syarat jika,

	Range	Nilai
Total Nilai	85 – 100	= A
Total Nilai	71 – 84	= B
Total Nilai	65 – 70	= C
Total Nilai	50 – 64	= D
Total Nilai	0 - 50	= E

Kalau diperhatikan pilihan ada 5 bagian, maka kondisi pengujian yang akan dilakukan adalah sebanyak jumlah pilihan – 1 karena pilihan terakhir tidak perlu diuji lagi artinya jika semua pilihan tidak dipenuhi berarti pilihan akan masuk ke pilihan terkahir.

```
if (Nilai > 85 && Nilai < 100)
    cout << "Nilai = A";
else if (Nilai > 71 && Nilai < = 85)
    cout << "Nilai = B";
else if Nilai > 65 && Nilai < = 70)
    cout << "Nilai = C";
else if Nilai > 50 && Nilai < = 64)
    cout << "Nilai = D";
else
    cout << "Nilai = E";
```

Contoh Program :

```
1   /* Program untuk if tunggal */
2
3   #include<constrea.h>
4   #include<conio.h>
5   void main()
6    {
7     int x,y;
8     x=10;
9     y=10 ;
10    if(x==y)
11    {
12     cout << "X sama dengan Y";
13    }
14     getch;
15    }
16    getch();
17  }
```

```
1   /*program untuk menguji bilangan positif dan negatif*/
2   #include<constrea.h>
3   #include<conio.h>
4   #include <string.h>
5   void main()
6    {
7                 long i;
8                 char str[80];
9
10                clrscr();
11                cprintf("Pembahasan if");
                  cprintf("Masukkan Angka Lalu tekan tombol
12  enter");
13                scanf("%d",&i);
14
15                If(i==0)
16                 {
```

```
17                    cprintf("Anda Memasukkan Angka Nol");
18                   }
19                   if(I>0)
20                   {
                      cprintf("Anda Memasukkan Bilangan
21        Positif");
22                   }
23                   if(I<0)
24                   {
                      cprintf("Anda Memasukkan Bilangan
25        Negatif");
26                   }
27
28                   cprintf("Tipe Data Kata");
29                   scanf("%s",&str);
30                   i=strlen(str);
31                   if(i)
32                   {
                      cprintf("Panjang Kata yang dimasukkan
33        adalah ", i);
34                   }
35                   if(I=1)
36                   {
37                    cprintf("i=1 selalu bernilai true\n"
38                   }
39
40                   getch;
41         }
```

```
1         /*program untuk menguji nilai Angka ke Huruf if3.cpp*/
2         #include <stdio.h>
3         #include <conio.h>
4
5         main()
6         {
7              int tot,q,f;
8
```

```
 9          clrscr();
10          cout <<  "Nilai Quiz =  " ); cin >> q ;
11          cout <<  "\n");
12          cout <<  "Nilai Final =  "; cin >> f ;
13          cout <<   "\n";
14          cout <<  "--------------------------------" ;
15          cout <<  "\n";
16
17          tot=0.3 * q + 0.7 * f;
18
19
20          if (tot >= 85 && tot <= 100 )
21            {
22              cout <<  "Nilai = A "  ;
23            }
24          Else
25          if (tot >= 75 && tot <= 84 )
26            {
27               cout <<   "Nilai = B " ;
28            }
29          Else
30          if (tot >= 65 && tot <= 74 )
31            {
32               cout <<  "Nilai = C " ;
33             }
34          Else
35          if (tot >= 55 && tot <= 64 )
36            {
37               cout <<   "Nilai = D "  ;
38             }
39          Else
40          if ( tot <= 55 )
41            {
42               cout <<   "Nilai = E " ;
43             }
44
45          getch() ;
46      }
```

```
1.  Contoh program menghitung akar-akar persamaan kuadrat
2.      #include <stdio.h>
3.      #include <math.h>
4.      void main()
5.      {
6.         int a,b,c,x1,x2;
7.         long D;
8.      printf("Masukkan nilai a : "); scanf("%i",&a);
9.      printf("Masukkan nilai b : "); scanf("%i",&b);
10.     printf("Masukkan nilai c : "); scanf("%i",&c);
11.        D=b*b-4*a*c;
12.        if (D==0) {
13.              x2=x1=-b/(2*a);
14.        printf("Akar-akar kembar. x1 = x2 = %i\n",x1);
15.     }
16.     else if (D>0) {
17.        printf("Akar-akar berlainan\n");
18.      x1=-b+sqrt(D);
19.      x2=-b-sqrt(D);
20.      printf("x1 = %i\n",x1);
21.      printf("x2 = %i\n",x2);
22.     }
23.     else printf("Akar imaginer\n");
24.     }
```

7.4 Struktur switch

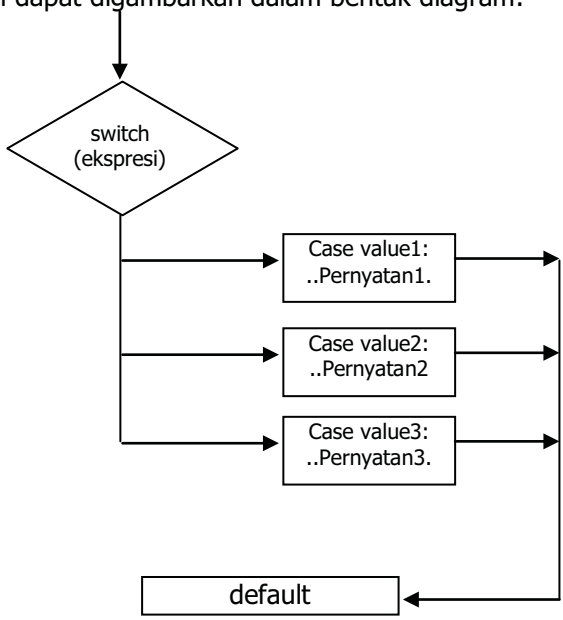
Bahasa C++ menyediakan sebuah struktur lain yang dapat digunakan untuk pengambilan keputusan n-jalur. Struktur ini bekerja jika jenis pilihan sama atau tidak multi kriteria. Dalam penggunaannya fungsi ini dapat digunakan jika pengujian yagn dilakukan lebih dari satu dan menguji sebuah nilai. Perintah **switch** secara umum dapat dituliskan sebagai berikut:

```
switch (ekspresi1)
   case value1:
          ……pernyataan1…
          break;
   case value2:
          ……pernyataan2…
          break;
   case value3:
          ……pernyataan3…
          break;
    ….
    default :
          ……pernyataanN…
```

penggunaan default adalah jika salah satu pilihan tidak dipenuhi maka nilai yang akan ditampilkan adalah nilai default.

Catatan:
Tuliskanlah kode program dengan rapi dan teratur agar mudah dbaca, diperbaiki serta mudah mendeteksi kesalahan.

Pernyataan ini dapat digambarkan dalam bentuk diagram:



Gambar Alur eksekusi Pernyataan switch

```
switch (x)
{
  Case 1:
         {
           printf(""Rp. 10000");
           break;
         }
  Case 2:
         {
           printf(""Rp. 15000");
           break;
         }
  default :
         {
           printf(""Rp. 20000")
          break;
         }
}
```

Contoh Program 1:

```
1   #include<stdio.h>
2   #include<conio.h>
3   #include<string.h>
4
5   main()
6    {
7     int pil;
8
9     clrscr();
10    cout <<  "Pembahasan switch – case \n";
11    cout << "\nMasukkan Pilihan Anda [1..5] : ";
12    cin >> pil;
13
14    switch (pil)
15    {
16     case  1:
17                cprintf( "\n Bilangan 1 ");
18                break;
19     case 2:
20                cprintf( "\n Bilangan 2 ");
21                break;
22     case  3:
23                cprintf( "\n Bilangan 3 ");
24                Break;
25     case  4:
26                cprintf( "\n Bilangan 4 ");
27                break;
28     case 5:
29                cprintf( "\n Bilangan 5 ");
30                break;
31     default:
32                cprintf( "\n Pilihan Tidak Ada");
33    }
34     getch();
35   }
```

Contoh Program 2 :

```
/* Program untuk perhitungan matematika  */

#include <iostream.h>
#include <stdio.h>
#include <conio.h>
void main()
{
  int pilihan;
  float Lsegitiga,Ksegitiga,Lpersegi,Kpersegi;
  float Lpersegipanjang,Kpersegipanjang,Llingkaran;
  float Klingkaran,Vkubus,LPkubus,Vbalok,LPbalok;
  float Vbola,LPbola,Vlimassegiempat,LPlimassegiempat;
  float Vtabung,LPtabung ;
  float a, t, s, p, l, r, phi,sAB,sBC,sCA;
  char ulang;
do
 {

  clrscr();

    cout<<"Menu Perhitungan "<<endl<<endl;
    cout<<"Menu Utama"<<endl;
    cout<<"1. Luas & Keliling Segitiga"<<endl;
    cout<<"2. Luas & Keliling Bujursangkar"<<endl;
    cout<<"3. Luas & Keliling Persegipanjang"<<endl;
    cout<<"4. Luas & Keliling Lingkaran"<<endl;
    cout<<"5. Isi & Luas Permukaan Kubus"<<endl;
    cout<<"6. Isi & Luas Permukaan Balok"<<endl;
    cout<<"7. Isi & Luas Permukaan Bola"<<endl;
    cout<<"8. Isi & Luas Permukaan Limas Segi Empat"<<endl;
    cout<<"9. Isi & Luas Permukaan Tabung"<<endl;
    cout<<"10. Exit"<<endl;
    cout<<endl<<endl;

    cout<<"Masukkan Pilihan anda : ";cin>>pilihan;
    switch (pilihan)
{
case 1 :
    cout<<"Menghitung Luas Segitiga"<<endl;
    cout<<"Menghitung Keliling Segitiga"<<endl;
    cout<<"Masukkan alas segitiga : ";cin>>a;
    cout<<"Masukkan tinggi segitiga : ";cin>>t;
    cout<<"Masukkan sisi AB segitiga : ";cin>>sAB;
```

```
40.         cout<<"Masukkan sisi BC segitiga : ";cin>>sBC;
41.         cout<<"Masukkan sisi CA segitiga : ";cin>>sCA;
42.         Lsegitiga = 0.5*a*t;
43.         Ksegitiga = sAB+sBC+sCA;
44.         cout<<"Luas segitiga adalah : "<<Lsegitiga<<endl;
45.         cout<<"Keliling segitiga adalah : "<<Ksegitiga<<endl;
46.         break;
47.     case 2 :
48.     cout<<"Menghitung Luas Bujursangkar"<<endl;
49.         cout<<"Menghitung Keliling Bujursangkar"<<endl;
50.         cout<<"Masukkan sisi persegi : ";cin>>s;
51.         Lpersegi = s*s;
52.         Kpersegi = 4*s;
53.         cout<<"Luas bujursangkar : "<<Lpersegi<<endl;
54.         cout<<"Keliling bujursangkar : "<<Kpersegi<<endl;
55.         break;
56.     case 3 :
57.         cout<<"Menghitung Luas Persegipanjang"<<endl;
58.         cout<<"Menghitung Keliling Persegipanjang"<<endl;
59.         cout<<"Masukkan panjang : ";cin>>p;
60.         cout<<"Masukkan lebar : ";cin>>l;
61.         Lpersegipanjang = p*l;
62.         Kpersegipanjang = (p+l)*2;
63.         cout<<"Luas Persegipnjng: "<<Lpersegipanjang<<endl;
64.         cout<<"Keliling Persegipnjng: "<<Kpersegipanjang<<endl;
65.         break;
66.     case 4 :
67.         cout<<"Menghitung Luas Lingkaran"<<endl;
68.         cout<<"Menghitung Keliling Lingkaran"<<endl;
69.         cout<<"Masukkan jari-jari lingkaran : ";cin>>r;
70.         phi = 3.14;
71.         Llingkaran = phi*r*r;
72.         Klingkaran = phi*(r+r);
73.         cout<<"Luas lingkaran adalah : "<<Llingkaran<<endl;
74.         cout<<"Keliling lingkaran adalah : "<<Klingkaran<<endl;
75.         break;
76.     case 5 :
77.         cout<<"Menghitung Volum Kubus"<<endl;
78.         cout<<"Menghitung Luas Permukaan Kubus"<<endl;
79.         cout<<"Masukkan sisi kubus : ";cin>>s;
80.         Vkubus = s*s*s;
81.         LPkubus = 6*s;
82.         cout<<"Volum kubus adalah : "<<Vkubus<<endl;
83.         cout<<"Luas permukaan kubus adalah : "<<LPkubus<<endl;
```

```
84.         break;
85.     case 6 :
86.         cout<<"Menghitung Volum Balok"<<endl;
87.         cout<<"Menghitung Luas Permukaan Balok"<<endl;
88.         cout<<"Masukkan panjang balok : ";cin>>p;
89.         cout<<"Masukkan lebar balok : ";cin>>l;
90.         cout<<"Masukkan tinggi balok : ";cin>>t;
91.         Vbalok = p*l*t;
92.         LPbalok = (2*p*l)+(2*p*t)+(2*l*t);
93.         cout<<"Volum balok adalah : "<<Vbalok<<endl;
94.         cout<<"Luas permukaan balok adalah : "<<LPbalok<<endl;
95.         break;
96.     case 7 :
97.         cout<<"Menghitung Volum Bola"<<endl;
98.         cout<<"Menghitung Luas Permukaan Bola"<<endl;
99.         cout<<"Masukkan jari jari bola : ";cin>>r;
100.        cout<<"Masukkan tinggi bola : ";cin>>t;
101.        phi = 3.14;
102.        Vbola = 4/3*phi*r*t*t*t;
103.        LPbola = 4*phi*r*r;
104.        cout<<"Volum bola adalah : "<<Vbola<<endl;
105.        cout<<"Luas permukaan bola adalah : "<<LPbola<<endl;
106.        break;
107.    case 8 :
108.        cout<<"Menghitung Volum Limas Segi Empat"<<endl;
109.        cout<<"Menghitung Luas Permukaan Limas Segi Empat"<<endl;
110.        cout<<"Masukkan panjang limas segi empat : ";cin>>p;
111.        cout<<"Masukkan lebar limas segi empat : ";cin>>l;
112.        cout<<"Masukkan tinggi limas segi empat : ";cin>>t;
113.        Vlimassegiempat = (p*l*t)*1/3;
114.        LPlimassegiempat = ((p+l)*t)+(p*l);
115.        cout<<"Volum limas segi empat adalah :
            "<<Vlimassegiempat<<endl;
116.        cout<<"Luas permukaan limas segi empat adalah :
            "<<LPlimassegiempat<<endl;
117.        break;
118.    case 9 :
119.            cout<<"Menghitung Volum Tabung"<<endl;
120.            cout<<"Menghitung Luas Permukaan Tabung"<<endl;
121.            cout<<"Masukkan jari jari tabung : ";cin>>r;
122.            cout<<"Masukkan tinggi tabung : ";cin>>t;
123.            phi = 3.14;
124.            Vtabung = phi*r*r*t;
125.            LPtabung = (2*phi*r)*(r*t);
```

```
126.            cout<<"Volum tabung adalah : "<<Vtabung<<endl;
127.            cout<<"Luas permukaan tabung adalah : "<<LPtabung<<endl;
128.            break;
129.    case 10 :
130.            cout<<"Keluar"<<endl;
131.            break;
132.    default:
133.            cout<<"Menu Tidak Ada"<<endl;
134.            break;
135.      }
136.          cout<<"Kembali ke Menu Utama (y/n)?";cin>>ulang;
137.     }
138.    while(ulang == 'y');
139.    }
```

Contoh Program untuk Kalkulator Sederhana

```
1.      #include <iostream.h>
2.      #include <conio.h>
3.      void main()
4.      {
5.      int op;
6.      int a,b,hsl;
7.      clrscr();
8.      cout << "\n KALKULATOR    ";
9.      cout << "\n 1. Tambah   ";
10.     cout << "\n 2. Kurang   ";
11.     cout << "\n 3. Kali   ";
12.     cout << "\n 4. Bagi   ";
13.     cout << "\n 5. Modulus  ";
14.     cout << "\n================================\n";
15.     cout << "Bilangan 1  : " ; cin >> a;
16.     cout << "Bilangan 2 : " ; cin >> b;
17.     cout << "PIlihan   : "; cin >>op;
18.     switch(op)
19.     {
20.     case 1:  { cout << "Hasil = " << a + b ;break;}
21.     case 2:  { cout << "Hasil = " << a - b ;break;}
22.     case 3:  { cout << "Hasil = " << a * b ;break;}
23.     case 4:  { cout << "Hasil = " << a / b ;break;}
24.     case 5:  { cout << "Hasil = " << a % b ;break;}
25.     default  : { cout << "Tidak ada Pilihan";}
26.     }
27.     getch();
28.     }
```

Contoh program 3 tentang rekursif

```
1.  #include <stdio.h>
2.  #include <conio.h>
3.  float pangkat(int A,int B)
4.  {
5.     if(B==0)
6.      {
7.       return 1;
8.      }
9.  else
10.     {
11.      return A* pangkat(A,B-1);
12.     }
13. }
14. int main()
15. { int A,B;
16. clrscr();
17. printf("masukan nilai yang akan dipangkatkan:");scanf("%d",&A);
18. printf("masukan pangkat:");scanf("%d",&B);
19. printf("%f",pangkat(A,B));
20. getch();
21. return 0;
22. }

23. Hasil
24. masukan nilai yang akan dipangkatkan: 2
25. masukan pangkat:5
    32
```

7.5 Struktur Goto

Struktur ini digunakan untuk memindahkan kendali eksekusi program dari satu bagian ke bagian lain dari sebuah fungsi tertentu. Bentuk umum adalah:

goto pengenal;

….

Label pernyataan;

Pengenal setelah pernyataan harus sama dengan label.

```
1    /* program untuk menggunakan fungsi goto */
2    #include<stdio.h>
```

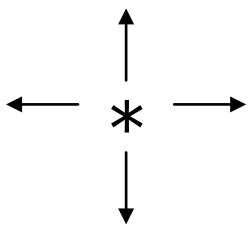
```
#include<conio.h>
#include<ctype.h>

int main(void)
 {
  unsigned char ch;

  clrscr();
  cprintf("Tuliskan sembarang tombol");
  while((ch =getchar()) != '\n')
    {
      // untuk menguji angka atau tidak
      if ((!isdigit(ch)) && (ch != '\n'))
          goto err_label;
          putchar(ch);
    }
    cprintf("\n");
    return 0;

    err_label:
    cprintf("\n\Errorr : %c bukan karakter angka \n",ch);
   getch();
 }
```

```
#include<stdio.h>
#include<conio.h>

main()
{
   char ch;

   int ax=1,ay=1;
   int x = 40, y=10;
   clrscr();
   _setcursortype(_NOCURSOR);

```

```
13          gotoxy(1,25);
14          do
15          {
16          gotoxy(x,y);
17           cprintf("*");
18          ch = getch();
19          If(ch==0)
20          {
21            ch = getch();
22            cgotoxy(x,y);
23            printf(" ");
24
25          switch(ch)
26           {
27             case 72 : if (y>=1);
28                                y=y-ay;
29                               textcolor(2);
30                               break;
31             case 80 : if (y<=25);
32                               y=y+ay;
33                               textcolor(3);
34                               break;
35             case 75 : if ( x>=1);
36                               x=x-ax;
37                               textcolor(4);
38                               break;
39             case 77 : if ( x<=80);
40                               x=x+ax;
41                               textcolor(5);
42                               break;
43          }
44
45                               if (x == 80 || x == 1)
46                                ax = -ax;
47
48                               if (y == 25 || y == 1)
49                                ay = -ay;
50
```

```
51          }
52         }
53          while (ch  != 27);
54       }
```

Output dapat digerakkan oleh panah ke atas, ke bawah , ke kanan dan ke kiri



BAB
8

PERULANGAN

Perulangan (*loop*) atau sering disebut dengan iterasi, adalah kumpulan instruksi program yang memerintahkan suatu tugas untuk dikerjakan secara berulang-ulang berdasarkan suatu kondisi tertentu/interval tertentu. Perulangan ini tentunya dapat dibatasi nilai atau jumlah perulangan yang dilakukan. Loop ini akan terus dilakukan selama kondisi yang dibandingkan oleh suatu loop tersebut masih memenuhi syarat atau bernilai true. Apabila sudah diluar syarat yang ditentukan maka perulangan akan berhenti atau bernilai false.

Jenis perulangan pada Bahasa C++:

- Perulangan for
- Perulangan while

8.1 Perulangan for

Perulangan dengan menggunakan statement f**or** sama halnya dengan melakukan perulangan while. Yang membedakan adalah bentuk penulisan serta menentukan nilai awal perulangan serta nilai akhir perulangan. Sehingga proses akan dilakukan selama interval nilai yang diberikan. Perulangan ini bisa ditentukan jumlah kelipatan kenaikan dari perulangan yang dilakukan.

Sintaks dari loop for adalah sebagai berikut:

```
for (ekspresi1;ekspresi2; ekspresi3)
  {
   ..pernyataan1…;
   ..pernyataan2…;
  }
```

Penjelasan:

Ekspresi1 : Nilai awal dari perulangan
Ekspresi2 : Nilai akhir dari suatu perulangan
Ekspresi3 : Besarnya kenaikan yang dilakukan selama perulangan
pernyatan : Instruksi-instruksi yang akan dilakukan jika kondisi memenuhi syarat

```
for (i=0; i<=10; i++)
   printf("%d\n",i);
```

Contoh progam:

```
1   #include<stdio.h>
2   #include<conio.h>
3
4   void main(void)
5    {
6     int i;
7     clrscr();
8     printf("Pembahasan LOOP \n");
9
10    for (i=1; i<6; i++)
11      printf("%d\n",i) ;
12
13     getch();
14   }
15
16  Output:
17  1
18  2
19  3
20  5
21  6
```

```
1   /* Program dengan penggunaan for*/
2
3   #include<stdio.h>
4   #include<conio.h>
5
6   main()
7   { //main
8    int i,a,z;
9    clrscr();
10    cprintf("Nilai Awal ");
11    scanf("%d",&a);
12    cprintf("Nilai Akhir");
13    scanf("%d",&z);
14
15    for (i=a; i<=z ; i++)
16      cprintf("%d \n",i) ;
17    getch() ;
18
19  } // end main
20
21  Output:
22  Nilai awal  : 2
23  Nilai akhir : 10
24  2
25  3
26  4
27  5
28  6
29  7
30  8
31  9
32  10
```

8.2 Perulangan while

Perulangan dengan menggunakan statemen while adalah loop yang melakukan pengulangan secara terus-menerus selama (*while*) suatu kondisi memenuhi syarat yang ditentukan kemudian jika tidak memenuhi akan berhenti.
Penulisan perintah ini diawali dengan while

Sintaksnya adalah sebagai berikut:

```
while  (ekspresi)
  {
        …pernyataan1 ..
        …Pernyataan 2..
  }
```

contoh

```
x = 1
while (x <= 6)
  {
    x++;
    printf("%d\n",x);
  }
```

Perintah yang terdapat dalam perulangan ini akan dikerjakan selama memenuhi syarat seperti contoh diatas. Program akan selalu menguji apakah nilai x lebih kecil sama dengan 6 maka jika lebih kecil kerjakan baris berikutnya untuk mencetak nilai x.

Pengertian program:

x = 1

while (x <= 6)

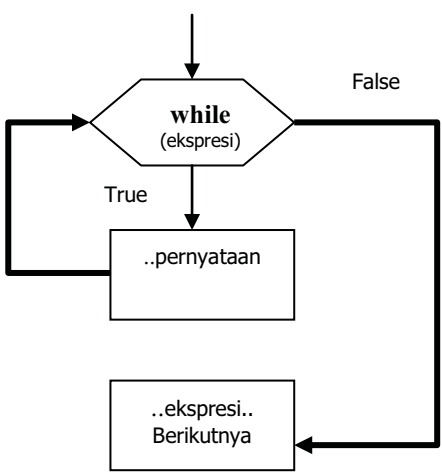
Variabel x akan diperiksa apakah isinya kurang atau sama dengan 6, jika benar maka akan dilajutkan

X++ → x = x + 1

variabel i dalam hal ini akan berubah-ubah setiap pengulangan dilakukan jika masih memenuhi persyaratan pada perulangan.

printf("%d\n",x);

Nilai dari variabel ini akan selalu dicetak setelah melakukan satu pengulangan



Gambar Alur eksekusi perulangan while

Contoh Progam 1:

```
 2
 3
 4      #include<stdio.h>
 5      #include<conio.h>
 6
 7      void main(void)
 8       {
 9        int i;
10        clrscr();
11        do
12        {
13          cprintf("Masukkan Bilangan [1..20] : ");
14          scanf("%d",&i);
15         }
16
17          while(i<=1 || i>=20);
18          //program ini akan dijalankan jika antara 1-20
19          {
20             cprintf("Angka yang diinput antara 1-20");
21          }
22         getch();
23       }
```

Contoh Program 2.

```
#include <constrea.h>
void main()
{
        clrscr();
        char A = '\0';
        do{
                A = getch();
                cout << "Hello" << endl;
                cout << "Belajar\n";
                cout << "A = " << A << endl;
                //getch();
        }while(A != 13);
}
```

8.3 Perulangan Bertingkat

Pengulangan bertingkat (*Nested Loop*) yaitu bahwa sebuah perulangan di dalam perulangan.

A. Sintaks dari loop for adalah sebagai berikut:

```
for (ekspresi1;ekspresi2; ekspresi3)
  {
        for (ekspresi1;ekspresi2; ekspresi3)
          {
                ..pernyataan1…
                ..pernyataan2…
          }
 }
```

b. Sintaks dari loop while adalah sebagai berikut:

```
  while  (ekspresi)
   {
        do
         {
                …pernyataan1 ..
                …Pernyataan 2..
         }
        while  (ekspresi)
   }
```

Contoh Programmya:

```
1     /* contoh program loop bersarang */
2     #include<stdio.h>
3     #include<conio.h>
4
5     void main()
6      {
7       int i,j;
8       clrscr();
9       cprintf("Contoh Loop Bersarang\n\n");
10
11      for (i=1; i<=3; i++)
12        {
13          cprintf("Bahasa Pemrograman C++\n");
14          for (j=1; j<=3; j++)
15           cprintf("    Mudah Dipelajari\n");
```

```
16          }
17
18          getch();
19        }
20
21        Output:
22        Bahasa Pemrograman C++
23              Mudah Dipelajari
24              Mudah Dipelajari
25              Mudah Dipelajari
26        Bahasa Pemrograman C++
27              Mudah Dipelajari
28              Mudah Dipelajari
29              Mudah Dipelajari
30        Bahasa Pemrograman C++
31              Mudah Dipelajari
32              Mudah Dipelajari
33              Mudah Dipelajari
```

```
 1      /* contoh program loop bersarang */
 2
 3      #include<constrea.h>
 4      #include<conio.h>
 5
 6      void main()
 7       {
 8        int i,j;
 9        clrscr();
10        cout << "Contoh Loop Bersarang\n\n";
11
12        for (i=1; i<=6; I+=1)
13          {
14            for (j=1; j<=i; j+=1)
15              cprintf("%d ",I*i);
16              cprintf("\n");
17          }
18
19          getch();
```

```
20      }
21
22

23      output :
24      1
25      4  4
26      9  9  9
27      16  16  16  16
28      25  25  25  25  25
29      36  36  36  36  36  36

1       /* contoh program loop bersarang segitiga bintang*/
2       #include<stdio.h>
3       #include<conio.h>
4       void main()
5       {
6        int i,j;
7        clrscr();
8        cout << "Contoh Loop Bersarang\n\n";
10       for (i=1; i<=6; i++)
11         {
12          for (j=1; j<=i; j++)
13            cprintf("*");
14            cprintf("\n");
15         }
16       //cetak kebawah
17       for (i=5; i>=1; i--)
18         {
19          for (j=1; j<=i; j++)
20            cprintf("*");
21            cprintf("\n");
22         }
23         getch();
24       }
25      Hasil :
26      *
```

```
27    * *
28    * * *
29    * * * *
30    * * * * *
31    * * * * * *
32    * * * * *
33    * * * *
34    * * *
35    *  *
36    *
```

```
 1    //untuk menentukan bilangan ganjil pada batas tertentu
 2    #include<constrea.h>
 3    #include<conio.h>
 4    #include<ctype.h>
 5
 6    void main()
 7    {
 8    int a,jlh,prt=0;
 9    char lg;
10    Do
11     {
12       clrscr();
13       cprintf(" Input Bilangan :"); scanf("%d",&a);
14       gotoxy(15,13);  // menentukan lokasi cetak teks di layar
15       cprintf(" Jlh Bilangan :"); scanf("%d",&jlh);
16
17       while(prt < jlh)
18        {
19         if(a % 2 == 1)
20          {
21           cprintf("%d ",a);
22           prt++;
23          }
24          A++;
25        }
26        cprintf("\n");
27        cprintf("Coba Lagi [Y/T] : ");
```

```
    lg=getche();
    printf("\n");
  }
  while(toupper(lg)=='Y');
   getch();
}

Output :
Input Bilangan :10
Jlh Bilangan : 5

11  13  15  17  19
          pada posisi baris 13 kolom 15
       Coba Lagi [Y/T] : y
```

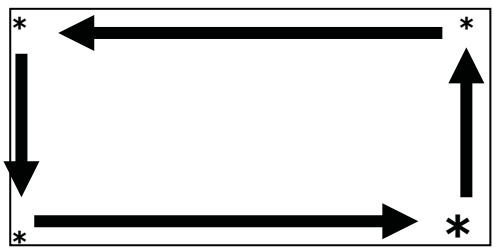
```
/* program  untuk menghitung bilangan prima */
#include <constrea.h>
#include <math.h>
#include <conio.h>

Void main()
{
 int i,j,ctr,awal,akhir;

 clrscr();
 cprintf("Berikan batas awal :" );scanf("%d",&awal);
 cprintf("Berikan batas akhir :" );scanf("%d",&akhir);
 cprintf("Bilangan Prima antara %d dan %d  = \n",awal,akhir);

 for (i=awal;i<=akhir;i++)
   {
    ctr = 0;
    for (j=1;j<=i;j++)
      {
       if (i % j == 0)
           ctr = ctr + 1;
      }
```

```
23          if (ctr == 2)
24            cprintf("%d     ",i);
25          }
26          getch();
27        }
29        Output:
30        Berikan Batas awal : 2
31        Berikan Batas Akhir : 10
32        Bilangan Prima antara 2 dan 10  =
33        2  3  5  7
 1        /* Proram untuk membuat huruf berjalan pada layar
 2           secara berkeliling mulai dari posisi sudut kiri ke bawah
 3           lanjut kekanan posisi sudat kanan bawah, naik ke
 4           sudut kanan atas dan mundur keposisi kiri atas*/
 5
 6        #include<constrea.h>
 7        #include<conio.h>
 8        #include<dos.h>  // fungsi watu dlm mili detik
 9
10        void main()
11        {
12          int y =1;
13          int x=1;
14         clrscr();
15         Do
16         {
17           // untuk menampilkan bintang dari line 1 sampai 24
18           // secara menurun
19           // y bertambah 1
20           Do
21           {
22             gotoxy(1,y);
23             cprintf("*");
24             delay (100);
25             gotoxy(1,y);
26             cprintf (" ");
27             Y++;
28           }
```

```
29          while(y<25 && !kbhit());
30
31          // untuk menampilkan bintang dari kolol 1 sampai 78
32          // secara mendatar
33          // x bertambah 1
34           do
35            {
36             gotoxy(x,24);
37             cprintf ( "*");
38             delay (100);
39             gotoxy(x,24);
40             cprintf ( " ");
41             X++;
42            }
43             while(x<79 && !kbhit());
45          // untuk menampilkan bintang dari line 24 sampai 1
46          // secara menaik
47          // y berkurang 1
48
49           Do
50            {
51             gotoxy(78,y);
52             cprintf ( " ");
53             y--;
54             delay (100);
55             gotoxy(78,y);
56             printf ( "*");
57             delay (100);
58            }
59          while(y>1 && !kbhit());
61          // untuk menampilkan bintang dari kolom 78 sampai 1
62          // secara mundur
63          // x berkurang 1
64           Do
65            {
66             gotoxy(x,1);
67             cprintf ( " ");
68             x--;
```

```
69            delay (100);
70            gotoxy(x,1);
71            cprintf ( "*");
72            delay (100);
73           }
74          while(x>1 && !kbhit());
75
76        }
77        while(!kbhit());
78       }
```

Output program huruf mengelilingi layar



/*Program Bilangan Rambang (*Random*)*/

```
1       #include <constrea.h>
2       #include <stdlib.h>
3       #include <conio.h>
4
5       int main(void)
6       {
7         int i;
8         randomize();
9         clrscr();
10        cout << "10 Bilangan random  0 to 99\n\n";
11        for(i=0; i<10; i++)
12            cprintf("%d\n", rand() % 10);
13        getch();
14        return 0;
15      }
16
17      output
18      10 Bilangan random  0 to 99
```

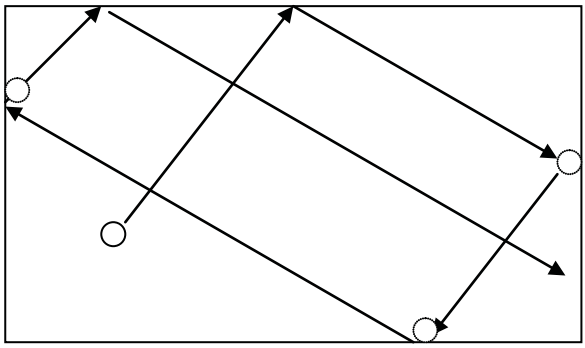
```
19      2
20      3
21      4
22      5
23      6
24      7
25      7
26      0
27      6
28      3
```

```
 1      /* Program bola pantul pantul.cpp*/
 2
 3      #include <constrea.h>
 4      #include <conio.h>
 5      #include <dos.h>
 6      #include <stdlib.h>
 7
 8      main()
 9      {
10       int x,y,dx,dy;
11       x=random(73)+3;
12       Clrscr();
13       y=random(21)+2;
14       dx=2*random(2)-1;
15       dy=2*random(2)-1;
16        Do
17        {
18         gotoxy(x,y);
19         cprintf(" ");
20         Y=y+dy;
21         X=x+dx;
22         if(y==1||y==25)
23         dy=-dy;
24          if(x==1||x==80)
25          dx=-dx;
26          gotoxy(x,y);
27          cprintf("*");
```

```
28          delay(100);
29
30      }
31      while(!kbhit());
32      _setcursortype(_NORMALCURSOR);
33      }
```

Output:

Bitang akan dipantulkan pada setiap sisi paling kiri, atas, bawah dan kanan. dan akan terus bergerak.



Gambar hasil program bola pantul

Program Bilangan Fibonacci

```
1   #include <constrea.h>
2   #include <conio.h>
3   int a1,a2;
4
5   void main(void)
6   {
7     int fibo(int);
8     int i,n;
9     clrscr();
10    cprintf("Masukkan jumlah bilangan Fibonacci  : ");
11    scanf("%d",&n);
12    cprintf("Deret %d Bilangan Fibonacci :\n",n);
13
14    for ( i=1;i<=n;++i)
15       cprintf("%d  ",fibo(i));
16       cprintf("\n");
17   getch();
18  }
19
20  int fibo(int a)
21  {
22    a = (a <=2) ? 1 : a1 + a2;
```

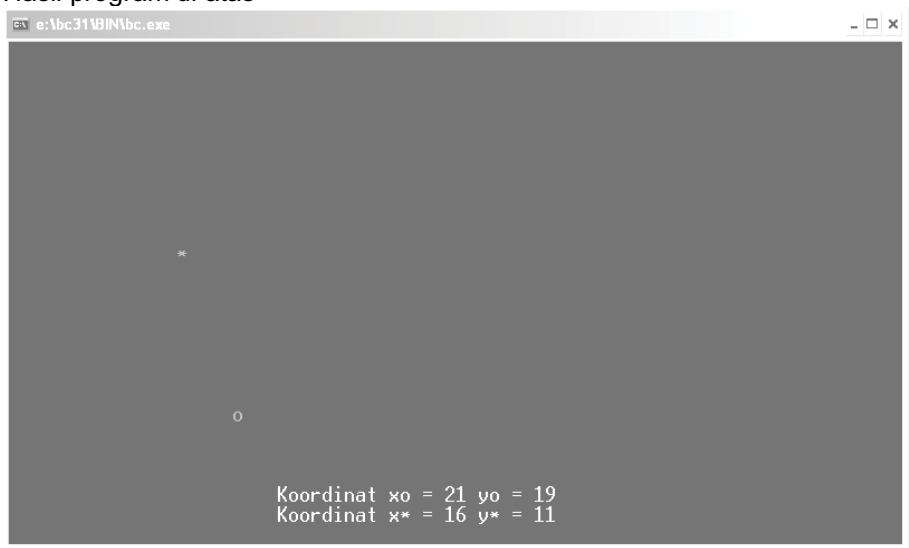
```
23        a2 = a1;
24        a1=a;
25
26        return a;
27      }
28
29      /* Pengertian Program
30      int a1,a2   : variabel eksternal dapat diakses dari seluruh program
31
32      for ( i=1;i<=n;++i)
33       cprintf("%d  ",fibo(i));
34
35      untuk melakukan perulangan dalam menampilkan bilangan fibonacci
36      sejumlah n yang dimasukkan
37
38      int fibo(int a)
39        a = (a <=2) ? 1 : a1 + a2;
40        a2 = a1;
41        a1=a;
42        return a;
43
44        Nilai yang akan dikirim ke fungsi diperiksa apakah lebih kecil
45        atau sama dengan 2, jika ya berikan nilai 1 ke a,
46        jika tidak masukkan nilai a1 + a2 ke a.
47        kemudian nilai a1 dipindahkan ke a2,
48        dan nilai a dipindahkan ke a1, Kemudian nilai a
49        dikembalikan ke fungsi utama main() sebagai suku bilangan fibonacci
50
51        OUTPUT:
52        Masukkan jumlah bilangan Fibonacci  : 10
53        Deret 10 Bilangan Fibonacci :
54        1 1 2 3 5 8 13 21 34 55
55        */
```

Program untuk Bola Pantul

```
1.  //PROGRAM UNTUK BOLA PANTUL
2.  #include <constrea.h>
3.  #include <stdlib.h>
4.  #include <dos.h>

5.  void main()
6.  {
7.  _setcursortype(_NOCURSOR);
8.  randomize();
9.  int x1, y1, ax1, ay1;
10. int x2, y2, ax2, ay2;
11. clrscr();
12.   x1 = random(50)+15;
13.   y1 = random(16)+5;
14.  ax1 = random(2)*2-1;
15.  ay1 = random(2)*2-1;
16.  x2 = random(50)+15;
17.  y2 = random(16)+5;
18.  ax2 = random(2)*2-1;
19.  ay2 = random(2)*2-1;
20. do
    {

        21   for(int i=1;i<16;i++)
        22   {
        23     textcolor(i);
        24     gotoxy(x1,y1);
        25     cprintf("o");
        26     gotoxy(x2,y2);
        27     cprintf("*");
        28     delay(20);
        29   }
        30    gotoxy(x1,y1);
        31   cprintf(" ");
        32   gotoxy(x2,y2);
        33   cprintf(" ");
        34   x1+=ax1;
        35   y1+=ay1;
        36   x2+=ax2;
        37   y2+=ay2;
        38   if(x1==15 || x1==65)
        39   ax1 = -ax1;
```

```
40    if(y1==5 || y1==20)
41      ay1 = -ay1;
42      if(x2==15 || x2==65)
43      ax2 = -ax2;
44      if(y2==5 || y2==20)
45      ay2 = -ay2;
46
47     gotoxy(25,23);
48      cout << "Koordinat xo = " << x1;
49      gotoxy(43,23);
50      cout << "yo = " << y1;
51      gotoxy(25,24);
52      cout << "Koordinat x* = " << x2;
53      gotoxy(43,24);
54      cout << "y* = " << y2;
55    }while(!kbhit());
56  }
```

Hasil program di atas



Contoh Program Faktorial

```
1.  #include <stdio.h>

2.  long Faktorial(int n)
3.  {
4.  long f=1;
5.  unsigned char i;

6.  for(i=2;i<=n;i++) f*=i;

7.  return f;
8.  }

9.  void main()
10. {
11. int nilai;
12. clrscr();
13. printf("Masukkan nilai yang akan dihitung faktorialnya : ");
14. scanf("%i",nilai);
15. printf("%i! = %ld\n",Faktorial(nilai));
16. }
```

Pada contoh di atas, output program yang akan dihasilkan jika dimasukkan nilai 5 adalah:

```
Masukkan nilai yang akan dihitung faktorialnya : 5
Hasil :
120
```

DAFTAR PUSTAKA

1. Aaron M. Tenebaum, Data Structures Using C and C++, Prentice Hall 1996
2. Abdul Kadir, Pemrograman C++, Andi Yogyakarta 2002
3. Darwis R. Manalu & Poltak Sihombing, Pemrograman Visual Basic, USU Pres 2011
4. Ekabrata, Algoritma & Pemrogarman Terstruktur dengan Micorosf Basic, PT Elex Media Computindo 1998
5. Robert L. Kruse, Data Structured & Program Design in C, PrenticeHalll, 1991
6. Yogianto H.M, Konsep Dasar Pemrograman Bahasa C, Andi Offset Yogyakarta, 1997

Drs. Humuntal Rumapea, M.Kom

Lahir di Pangururan 16 Agustus 1962 adalah lulusan Jurusan Matematika FMIPA Universitas Sumatera Utara tahun 1987 dan Magister Ilmu Komputer FASILKOM Universitas Indonesia tahun 2001. Saat ini bekerja sebagai Dosen Tetap di Fakultas Ilmu Komputer Universitas Methodist Indonesia, dan telah tersertifikasi sebagai dosen Profesional untuk mata kuliah Algoritma dan Pemrograman, Pengolahan Citra, serta Logika Fuzzy.

Disamping sebagai penulis buku juga memiliki pengalaman manajemen sebagai Ketua Program Studi, wakil Dekan I bidang Akademik dan sejak tahun 2011 sebagai Dekan Fakultas Ilmu Komputer Universitas Methodist Indonesia.

Darwis R. Manalu, S.Kom,M.M

Kelahiran Nagurguran 10 April 1976 Kabupaten Humbang Hasundutan adalah lulusan Teknik Informatika. Adalah seorang Praktisi Teknologi Informasi bidang Pemrograman dan Sistem Analist dan telah menekuni penulisan buku-buku komputer sejak tahun 1998. Sekarang ini bekerja sebagai Dosen Kopertis Wilayah I Diperbantukan pada Fakultas Ilmu Komputer Universitas Methodist Indonesia Medan, dan mengajar di Universitas Sumatera Utara dan STMIK Mikroskil. Dengan mengampu Pemrograman Sistem Informasi, Analisa Perancangan Sistem, Manajemen Proyek Teknologi Informasi dan Analisa Proses Bisnis. Beberapa buku yang telah ditulis adalah Pemrograman Visual Basic, Pemrograman Multiuser.

Dengan Mempelajari Buku Ini Diharapkan:

1. Mampu mengoperasikan C++
2. Menguasai Teknik Pemrograman C++
3. Dapat Menguasai Algoritma dan Pemrogarman
4. mamapu Menyelesaikan Permasalahan Dengan C++
5. Mampu mengembangkan sendiri

Buku Berisi Tentang:

- Algoritma dan Pemrograman
- Pemrograman Bahasa
- Penulisan C++
- Jenis Data
- Operator Dan Ekspresi
- Fungsi Input Dan Output
- Seleksi (Selection)
- Perulangan (Looping)
- Pointer
- Fungsi (Function)
- Larik (Array)
- Fungsi String
- Struktur (Structure)
- File Input/Output
- Class dan Objek

ISBN 979-458-662-5

usupress.usu.ac.id